

LIETUVOS EDUKOLOGIJOS UNIVERSITETAS

Violeta Jadzgevičienė

**MOKYMOŠI STILIŲ IR MOKYMOŠI
METODŲ DERMĖ KAIP STUDENTŲ
PROGRAMAVIMO GEBĖJIMŲ UGDYMOŠI
VEIKSNYS**

DAKTARO DISERTACIJA

Socialiniai mokslai, edukologija (07 S)

*Lietuvos
edukologijos
universiteto
leidykla* 

Vilnius, 2015

Disertacija rengta 2008–2015 metais Lietuvos edukologijos universitete.

Mokslinės vadovės:

prof. dr. Joana Lipeikienė, 2008–2010 (Lietuvos edukologijos universitetas, socialiniai mokslai, edukologija – 07 S)

prof. dr. Valentina Dagienė, 2011–2015 (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07 T)

© Violeta Jadzgevičienė, 2015

© Lietuvos edukologijos universiteto leidykla, 2015

TURINYS

ĮVADAS	5
Disertacijoje vartojamos sąvokos	13
Disertacijoje vartojamos santrumpos	15
1. MOKYMO SI STILIŲ TIPOLOGIJŲ NAUDOJIMO UGDYMO SI PROCESI IR STUDENTŲ PROGRAMAVIMO GEBĖJIMŲ UGDYMO SI TEORINĖS PRIELAI DOS	16
1.1. Mokymosi stilių tipologijų naudojimo aspektai.....	16
1.2. Pradedančiųjų programavimo gebėjimų ugdymosi problematika	25
1.3. Pradedančiųjų programavimo gebėjimams ugdytis taikomi mokymosi metodai ..	38
2. STUDENTŲ PROGRAMAVIMO GEBĖJIMŲ UGDYMO SI, ĮVERTINANT MOKYMO SI STILIŲ IR MOKYMO SI METODŲ DERMĖ, TYRIMO ORGANIZAVIMAS IR METODOLOGIJOS PAGRINDIMAS	46
2.1. Tyrimo organizavimas ir atlikimas	47
2.2. Mokymosi stilių ir mokymosi metodų dermės modelis pradedančiųjų programavimo gebėjimams ugdytis	49
2.3. Žvalgomojo tyrimo metodika	58
2.4. Veiklos tyrimo metodika	65
2.5. Tyrimo etika	73
2.6. Tyrimo apribojimai	74
3. STUDENTŲ PROGRAMAVIMO GEBĖJIMŲ UGDYMO SI, ĮVERTINANT MOKYMO SI STILIŲ IR MOKYMO SI METODŲ DERMĖ, TYRIMO EMPIRINIS PAGRINDIMAS	76
3.1. Žvalgomojo tyrimo rezultatai	76
3.1.1. Programavimo gebėjimų ugdymo programos Lietuvos aukštosiose mokyklose.....	76
3.1.2. Kiekybinis tyrimas „Studentų požiūris į įvadinio programavimo mokymąsi“	87
3.1.3. Kokybinis tyrimas „Įvadinio programavimo dėstymo strategijos“	99
3.1.4. Žvalgomojo tyrimo rezultatų apibendrinimas.....	102
3.2. Mokymosi stilių ir mokymosi metodų dermės modelio tyrimas	105
3.2.1. Mokymosi metodų vertinimai	105
3.2.2. Studentų požiūris į taikytus mokymosi metodus.....	111

3.2.3. Veiklos tyrimo rezultatų apibendrinimas.....	133
TYRIMO REZULTATŲ APTARIMAS IR DISKUSIJA.....	135
IŠVADOS.....	137
PASIŪLYMAI IR PRAKTINĖS REKOMENDACIJOS.....	139
NAUDOTA LITERATŪRA.....	140
PRIEDAI.....	152

ĮVADAS

Temos aktualumas

Šiuo metu visa Lietuvos švietimo sistema išgyvena paradigimų kaitą, kuomet nuo mokymo paradigmos pereinama prie mokymosi paradigmos. Mokymas yra neatsiejamas nuo mokymosi – tai ne du vienas šalia kito egzistuojantys dalykai, o organiškai jų visuma. Tik vykstant mokymosi procesui, galima kalbėti apie mokymo procesą. Mokymosi paradigmą atliepia ir studentus orientuotos studijos – tai tam tikras požiūris ir aukštojo mokslo sistema, kuris grindžiamas studijų rezultatais ir Europos kreditų kaupimo bei perkėlimo sistema. Vadovaujantis šiuo požiūriu besimokantysis ir mokantysis skatinami glaudžiau bendradarbiauti, akcentuojamas besimokančiojo savarankiškumas ir jo gebėjimas įsitraukti į mokymosi procesą. Mokymas ir mokymasis turi atitikti nuolat kintančius besimokančiųjų poreikius, gebėjimus, polinkius, turimą patirtį, interesus. Ugdymosi procese naudojamos įvairios technologijos (kompiuteriai, išmanieji ir mobilieji įrenginiai, interneto išteklių, virtualiosios mokymosi aplinkos, socialiniai tinklai ir kt.), kuriamos įvairios adaptyvios elektroninės mokymosi aplinkos, galinčios prisitaikyti prie besimokančiojo poreikių. Kintant mokymosi aplinkai, keičiasi ir mokymosi proceso planavimas, jo organizavimas ir eiga. Tai kelia naujus iššūkius pedagogams. Kadangi mokymasis vyksta visur – ir realioje, ir virtualiojoje erdvėje, būtina atlikti įvairiapusių tyrimus, apimančius mišrųjų mokymąsi (ir klasėje, ir virtualiojoje erdvėje), naudojantis įvairiomis informacinių ir komunikacinių technologijų teikiamomis galimybėmis. Elektroninio mokymosi tyrėjai Lietuvoje akcentuoja edukologinių tyrimų svarbą. Technologinių tyrimų tikslas – projektuoti, įgyvendinti, parinkti, vertinti ir panašią programinę įrangą, skirtą edukaciniam procesui. Kita vertus, edukaciniame procese, be technologinių problemų, iškyla ir daugybė socialinių edukologinių, į kurias taip pat būtina atsižvelgti (Dagienė, Žilinskienė, 2011; Beres ir kt., 2012). Veikiamas nuolatinės informacijos ir komunikacijos technologijų kaitos, programavimo mokymasis taip pat nuolat kinta. Studijoms naudojant virtualiąsias mokymosi aplinkas derinamas pedagogo organizuojamas mokymasis su mokymusi kompiuterių tinkle – tai skatina studentus nuolat savarankiškai mokytis (Garnham, Kaleta, 2002).

Lietuvos aukštosiose mokyklose yra parengta ir įgyvendinama daugiau kaip 70 pirmosios pakopos universitetinių ir koleginių studijų programų, pagal kurias rengiami dauguma būsimųjų informatikos ir komunikacijos technologijų (IKT) specialistų. Didžioji dalis studentų pirmaisiais studijų metais studijuoja kursą, skirtą programavi-

mo kompetencijai ugdyti. Kadangi šis kursas studijų programose vadinamas įvairiai, disertaciniame darbe jis įvardijamas įvadinio programavimo kursu (ĮPK). Pradedant mokytis programuoti, pirmiausia susiduriama su problemų sprendimu. Problemos sprendimą reikia išreikšti taip, kad jį „suprastų“ kompiuteris, todėl tenka mokytis programavimo kalbos sintaksės ir semantikos. Mokydamiesi studentai susikuria „savą“ supratimą ir skirtingai interpretuoja mokomąją medžiagą. Kaip teigia L. Mannila (2009), taip suformuojami skirtingi mentaliniai modeliai, tačiau neteisingi modeliai sukelia įvairių mokymosi sunkumų (Mannila, 2009). Nemaža dalis pradedančiųjų programuotojų patiria tokių programavimo mokymosi sunkumų. Išvardysime keletą svarbiausių priežasčių, sukeliančių šiuos sunkumus:

- būtina turėti įvairių įgūdžių: analizės, projektavimo, loginio mąstymo, kartotinio naudojimo, vertinimo, refleksijos,
- norint mokytis programavimo svarbu įgyti kompiuterio veikimo principų žinių, o tai nėra lengva.

Mokslinėje literatūroje (du Boulay, 1989; Gomes, Mendes, 2007; Bennedsen, 2008; Urbonienė, 2014) pateikiama programavimo mokymosi sunkumus lemiančių priežasčių klasifikacija:

- netinkami mokymo metodai (programavimo mokymas nėra individualizuotas, taikomi metodai nedera su besimokančiųjų mokymosi stiliais, dažnai dinaminių konceptų mokoma naudojant statinę medžiagą, susitelkiama į programavimo kalbos ir jos sintaksės mokymą, o ne į uždavinių sprendimą naudojant programavimo kalbą ir aplinką),
- netinkami mokymosi metodai (besimokantieji taiko netinkamus mokymosi metodus, nepakankamai dirba savarankiškai),
- nepakankami besimokančiųjų gebėjimai ir neigiamos nuostatos (neteisingai transformuojama natūralios kalbos semantika, besimokantieji turi būti įgiję arba norėti įgyti įvairių su programų kūrimu susijusių gebėjimų ir pan.),
- sudėtinga programavimo prigimtis (programavimas reikalauja aukšto abstrakcijos lygmens, programavimo kalbų sintaksė yra sudėtinga),
- psichologinės priežastys (silpna motyvacija, programavimo pradedama mokytis paaužlystėje ar tampant suaugusiuoju, o tai sudėtingas amžiaus tarpsnis).

Išsamiai aiškinant patiriamus sunkumus mokymo metodų aspektu, pabrėžiama, kad programavimo mokymas nėra individualizuotas, o pedagogo taikomi metodai nedera su besimokančiųjų mokymosi stiliais (Gomes, Mendes, 2007; Urbonienė, 2014). Pradedančiųjų patiriami programavimo mokymosi sunkumai gali tapti ir studijų nutraukimo priežastimi. Studijų nutraukimo veiksniai IKT krypties studijose

Lietuvos mastu nebuvo tyrinėti, tačiau žinoma, kad IKT studijas pasiseka baigti tik pusei įstojusiųjų (Plečkaitis, Žvirblis, 2013).

Informatikos mokymosi moksliniai tyrimai yra neseniai susiformavusi mokslinių tyrimų kryptis, įtraukianti informatikos, informatikos inžinerijos, psichologijos ir edukologijos mokslininkus. Parengta disertacija priskiriama programavimo mokymosi tyrimams – tai atskira informatikos mokymosi tyrimų kryptis.

Programavimo gebėjimai ir jų ugdymas nagrinėti daugelio užsienio mokslininkų (Van-Roy, Haridi, 2002; de Raadt, Watson, Toleman, 2003; Bruce, 2004; Bennedsen, Caspersen, 2004; Vujošević-Janičić, Tošić, 2008 ir kt.) darbuose. Lietuvoje atlikta vos keletas tokių tyrimų (Dagienė, Urbonienė, 2010 ir kt.). Daug dėmesio skiriama probleminiam klausimui, nuo ko pradėti programavimo gebėjimų ugdymą: nuo duomenų struktūrų, algoritmų, loginio programavimo, kompiuterių techninės įrangos ar konkrečios programavimo kalbos? Apibendrinus atliktos mokslinės literatūros analizės rezultatus, nustatytos tokios programavimo mokymosi tyrimų kryptys:

- programavimo paradigmos, programavimo kalbos ir programavimo aplinkos pasirinkimas (Hadjerrouit, 1998, 2008; Ben-Ari, 2001; Barnes, Kölling, 2002; Vujošević-Janičić, Tošić, 2008; Mason, Cooper, de Raadt, 2012; Dagienė, Urbonienė, 2010; Maknickas, 2012; Urbonienė, 2012; Farag, Ali, Deb, 2013; Waleed, Sanwar, Debzani, 2013),
- programavimo mokymo ir mokymosi metodų pasirinkimas (Barnes Fincher, Thompson, 1997; Dagdilelis, 2004; Gonzales, 2004; Bennedsen, Caspersen, 2004; Sajaniemi, Kuittinen, 2005; Wulf, 2005; Caspersen, 2007; Vidžiūnas, Mickus, 2007; Lopez, Whalley, Robbins, Lister, 2008; Bennedsen, 2008; Sorva, 2010; Vargas, Jimenez, Santamaria, Navarro, Aviles, 2011; Zacharis, 2011; Vitkutė-Adžgauskienė, Vidžiūnas, 2012; O’Kelly, Mooney, Ghent, Gaughran, Dunne, Bergin, 2012),
- įvairių šiuolaikinių mokymo technologijų, gerinančių programavimo mokymą, kūrimas ir naudojimas (Gibbs, 2000; Garnham, Kaleta, 2002; Van Roy, Haridi, 2002, 2004; Lipeikienė, 2004; Yong, 2007; Theodorou, Kordaki, 2010; Dagienė, Urbonienė, 2010; Juškevičienė, Urbonienė, 2011; Sorva, 2012; Jadzgevičienė, Urbonienė, 2013; Urbonienė, 2013; Bennane, 2013; Guo, 2013; Tillmann, Halleux, Tao, Gulwani, Bishop, 2013),
- studentų motyvacija mokantis programuoti (Kinnunen, 2009; Berglund, Lister, 2010),

- programavimo mokymosi sėkmių ir nesėkmių priežastys (du Boulay, 1989; Jenkins, 2002; Robins, Rountree, Rountree, 2003; Gomes, Mendes, 2007; Lister, Adams, Fitzgerald ir kt., 2004; Mannila, 2007; Kölling, 2012).

Nors Lietuva turi puikias programavimo mokymo tradicijas ir patirtį, yra parengta nemažai mokymo priemonių ir didaktinės medžiagos, tačiau galima teigti, kad programavimo mokymosi tyrimų, kuriuose būtų analizuojama programavimo gebėjimų ugdymas, mokymosi metodų pasirinkimas, studentų motyvacija mokytis programuoti, mokymosi sėkmių ir nesėkmių priežastys, labai trūksta. Nuolat kintantis programavimo ugdymo turinys ir didelis studijas nutraukiančių studentų skaičius skatina ieškoti veiksmingesnių mokymo bei mokymosi priemonių, metodų ir būdų.

Renkantis disertacinio tyrimo kryptį, atkreiptas dėmesys į mokymosi stilių ir mokymosi metodų dermę. Egzistuoja daug ir įvairių mokymosi stilių teorijų ir modelių, kuriuose išskiriami skirtingi mokymosi stilių tipai. Teigiama, kad egzistuoja tiesioginis ryšys tarp mokymosi stiliaus, mokymosi būdų ir mokymosi rezultatų (Schmeck, 1988; Biggs, 2003; Biggs, Tang, 2007). Mokslininkų teigimu, kuo tiksliau nustatomas ryšys tarp besimokančiojo mokymosi stiliaus ir mokymosi veiklos, tuo veiksmingesnis yra mokymasis (Jensen, 2001; Zacharis, 2011; Preidys, Žilinskienė, 2012). Mat, atsižvelgiant į mokymosi stiliui būdingas ypatybes, galima planuoti mokymosi veiklas, parinkti besimokančiajam tinkamus mokymosi metodus. Apibendrinant poveikio mokymosi rezultatams tyrimus, kai ugdant atsižvelgiama į besimokančiųjų mokymosi stilių, dažniausiai laikomasi prielaidos, kad tam tikrą mokymosi stilių atitinkantys besimokantieji yra panašūs ir turintys tokių pačių poreikių. Tačiau šiuo atveju lieka neįvertinti individualūs besimokančiųjų skirtumai. Nors moksliniuose straipsniuose randama teorinių ir empirinių įrodymų, kad ugdymas, pagrįstas mokymosi stiliais, padeda mokytis, stiprina motyvaciją, tačiau yra ir tyrimų, prieštaraujančių teiginiui, kad studentas išmoksta geriau, kai yra mokomas jo mokymosi stilių atitinkančiais metodais (Pashler, McDaniel, Rohrer, Bjork, 2008). Egzistuojančios skirtingos nuomonės skatina išsamiau tyrinėti mokymosi stilių ir mokymosi metodų dermę.

Tyrimo problema

Kaip ugdomi studentų programavimo gebėjimai aukštosiose mokyklose derinant studentų mokymosi stilius su taikomais mokymosi metodais?

Tyrimo objektas

Studentų mokymosi stilių ir mokymosi metodų dermė ugdant programavimo gebėjimus.

Tyrimo tikslas

Atskleisti studentų programavimo gebėjimų ugdymosi ypatumus į studentus orientuoto ugdymo kontekste.

Tyrimo uždaviniai

1. Išanalizuoti mokslinę literatūrą, švietimo dokumentus ir atskleisti teorines mokymosi stilių taikymo prielaidas ugdant programavimo gebėjimus.
2. Sudaryti mokymosi stilių ir mokymosi metodų dermės (MSMMD) modelį, skirtą programavimo gebėjimams ugdyti.
3. Numatyti MSMMD modelio taikymo galimybes pradedančiųjų programavimo gebėjimams ugdyti.
4. Remiantis MSMMD modeliu atskleisti mokymosi stilių ir mokymosi metodų dermės ypatumus pradedančiųjų programavimo gebėjimų ugdymosi procese.

Ginami teiginiai

1. Sklandžią pradedančiųjų programavimo proceso eigą lemia loginio mąstymo, problemų sprendimo, programavimo konceptų suvokimo, programavimo kalbos sintaksės ir semantikos, informacijos paieškos ir analizės, programinių fragmentų derinimo, programavimo aplinkos naudojimo, programavimo technikos ir metodų taikymo gebėjimų visuma, kuri ir gali būti įvardijama kaip pradedančiųjų programavimo gebėjimas.
2. Įvertinus mokymosi stilių ir mokymosi metodų dermės veiksnį galima įveikti programavimo mokymosi pradžioje kylančius sunkumus, susijusius su mokymosi metodų pasirinkimu ir taikymu, įvairiau ir tikslingiau planuoti ugdomąją veiklą. Sukurtas mokymosi stilių ir mokymosi metodų dermės modelis tinkamas ugdant pradedančiųjų programavimo gebėjimus ir gali būti taikomas individualizuojant ugdymą, tobulinant mokėjimo mokytis kompetenciją arba kuriant adaptyvias ir rekomendacines programavimo mokymosi aplinkas. Kalbant apie vizualiuosius metodus, tai pasirinktos mokymosi stilių tipologijos tinkamumas nepagrįstas.

Taikyti tyrimo ir duomenų analizės metodai

Atliekant disertacinį tyrimą taikyti įvairūs tyrimo ir duomenų analizės metodai:

- *mokslinės literatūros, edukologinių, psichologinių tyrimų sisteminė analizė* naudota tiriant: 1) mokymosi stilių naudojimo pedagoginėje praktikoje as-

- pektus; 2) pradedančiųjų programavimo gebėjimų ugdymosi problematiką; 3) programavimo gebėjimams ugdytis taikomus mokymosi metodus;
- *dokumentų turinio analizė* naudota tiriant, kaip pradedančiųjų programavimo gebėjimų ugdymas įgyvendinamas Lietuvos aukštųjų mokyklų (universitetų ir kolegijų) įvadinio programavimo kursuose;
 - *anketinė apklausa* naudota siekiant išsiaiškinti studentų turimą patirtį ir pasirengimą programavimo studijoms, jų požiūrį į programavimo mokymąsi, naudojamus ir pageidaujamus mokymosi būdus;
 - *modeliavimas* – tai mokymosi stilių ir mokymosi metodų dermės modelio kūrimas;
 - *veiklos tyrimas* naudotas taikant mokymosi stilių ir mokymosi metodų dermės modelį ir vertinant jo tinkamumą ugdantis programavimo gebėjimus;
 - *interviu* naudotas siekiant išsiaiškinti Lietuvos aukštosiose mokyklose ĮPK dėstytojų naudojamas programavimo gebėjimų ugdymo strategijas ir studentų požiūrį į šias strategijas, atliekant mokslinį tyrimą;
 - *ugdymųjų situacijų stebėjimu* siekta užfiksuoti studentų nuostatas, elgseną ir motyvaciją, kai atliekamos individualaus ir grupinio darbo veiklos;
 - *turinio analizė* naudota analizuojant studentų ir dėstytojų interviu duomenis;
 - *matematinė statistika* naudota vertinant klausimynų patikimumą ir atliekant kiekybinio tyrimo duomenų analizę.

Tyrimo rezultatų mokslinis naujumas ir teorinis reikšmingumas

Išanalizavus mokslinę literatūrą ir dokumentus teoriškai pagrįsta pradedančiųjų programavimo gebėjimų ir jų ugdymosi samprata. Remiantis užsienio mokslininkų pasiūlytu modeliu disertaciniame darbe buvo sukurtas mokymosi stilių ir mokymosi metodų dermės modelis, skirtas pradedančiųjų programavimo gebėjimams ugdyti. Modelis pagrįstas P. Honey ir A. Mumfordo mokymosi stilių tipologija, įtraukiant programavimo gebėjimams ugdytis tinkamus mokymosi metodus. Taip patobulinus modelį sudarytos prielaidos juo pasinaudoti ugdant pradedančiųjų programavimo gebėjimus. Teorinės įžvalgos ir empirinis tyrimas papildė pradedančiųjų programavimo gebėjimų ugdymosi didaktiką naujomis išvadomis apie programavimo gebėjimų ugdymąsi.

Tyrimo praktinis reikšmingumas ir taikymo perspektyvos

Pradedančiųjų programavimo gebėjimams ugdytis pritaikyta P. Honey ir A. Mumfordo mokymosi stilių tipologija. Sukurtas mokymosi stilių ir mokymosi metodų

dermės modelis taikytinas projektuojant adaptyvias ar rekomendacines programavimo mokymosi aplinkas. Parengta pradedančiųjų programavimo gebėjimų ugdymosi programa, ištirti programavimo gebėjimams ugdytis taikomi mokymosi metodai ir pagrįstas jų tinkamumas skirtingiems mokymosi stiliams. P. Honey ir A. Mumfordo mokymosi stilių nustatymo klausimynas išverstas, apčiuotas ir adaptuotas besimokantiems programuoti.

Tyrimo rezultatų mokslinis aprobavimas

1. Publikacijos recenzuojamuose periodiniuose moksliniuose leidiniuose:

1.1. Čechamirienė, G., Jadzgevičienė, V., Stankevičienė, E. (2014). Pradedančiųjų programavimo gebėjimų ugdymas. *Mokslas ir edukaciniai procesai*, 1 (18), 21–29. ISSN 2345-0681.

1.2. Jadzgevičienė, V., Urbonienė, J. (2013). The Possibilities of Virtual Learning Environment Tool Usability for Programming Training. *Innovative Infotechnologies for Science, Business and Education*, 1 (14), 3–9. ISSN 2029-1035.

1.3. Jadzgevičienė, V. (2013). Programavimo pradmenys Lietuvos kolegijose. *Vadyba*, 2 (23), 133–139. ISSN 1648-7974.

1.4. Jadzgevičienė, V., Urbonienė, J. (2012). *Cooperation in Programming Learning*. Proceedings of the 12th Koli Calling International Conference on Computing Education Research, 143–144. ISBN 978-1-4503-1795-5.

1.5. Valavičius, E., Jadzgevičienė, V. (2012). New Trends in Informatics Study Programmes. *Innovative Infotechnologies for Science, Business and Education*, 1 (12), 14–16. ISSN 2029-1035

2. Pranešimai mokslinėse konferencijose:

2.1. Jadzgevičienė, V., Urbonienė, J. (2013). The Possibilities of Virtual Learning Environment Tool Usability for Programming Training. In *The 6th International Conference on Innovative Infotechnologies for Science, Business and Education (IIT – 2013)*. Vilnius.

2.2. Jadzgevičienė, V. (2013). Programavimo pradmenys Lietuvos kolegijose. Iš *X tarptautinė mokslinė-metodinė konferencija „Regioninės problemos: ekonomika, vadyba, technologijos“*. Klaipėda.

2.3. Jadzgevičienė, V. (2013). Įvadinio programavimo dalykas informatikos ir informatikos inžinerijos krypčių profesinio bakalauro studijų programose. Iš *XVI tarptautinė mokslinė kompiuterininkų konferencija*. Šiauliai.

2.4. Jadzgevičienė, V. (2012). Programavimo mokymo metodų ir studentų mokymosi stiliaus sąsajos. (Learning Styles of University Students in Relation to Programming

Teaching Methods). Iš *III tarptautinis doktorantų seminaras: Informatikos ir informatikos inžinerijos mokymo tyrimai: metodologija, metodai ir praktika*. Druskininkai.

2.5. Jadzgevičienė, V., Urbonienė, J. (2012). Bendradarbiavimas mokantis programuoti (Cooperation in Programming Learning). In *The 12th Koli Calling International Conference on Computing Education Research*. Suomija.

2.6. Valavičius, E., Jadzgevičienė, V. (2012). Naujos tendencijos informatikos krypties studijų programose (New Trends in Informatics Study Programmes). Iš *5-oji tarptautinė konferencija „Inovatyvios informacinės technologijos mokslui, verslui ir švietimui“*. Vilnius.

Disertacijos struktūra ir apimtis

Darbą sudaro: terminų ir santrumpų žodynas, teorinės tiriamos problemos įžvalgos, disertacinio tyrimo metodologija, disertacinio tyrimo rezultatai, gautų rezultatų aptarimas, išvados ir rekomendacijos, naudotos literatūros sąrašas ir priedai. Darbo apimtis yra 151 puslapis be priedų. Tekste panaudota 26 paveikslai, 39 lentelės ir 11 priedų. Rašant disertaciją panaudoti 229 literatūros šaltiniai.

Įvade pristatomas darbo aktualumas ir problematika, darbo tikslai ir uždaviniai, mokslinis naujumas ir teorinis reikšmingumas, praktinė darbo reikšmė ir darbo aprobavimas. Pirmajame skyriuje atskleidžiama mokymosi stilių taikymo ugdymosi procese ir programavimo gebėjimų ugdymosi teorinės prielaidos. Antrajame skyriuje aprašoma disertacinio tyrimo metodologija, pateikiama disertacinio tyrimo schema, atskirų tyrimo dalių metodika, mokslinio tyrimo apribojimai, tyrimo etika. Trečiajame skyriuje pateikiami disertacinio tyrimo rezultatai.

DISERTACIJOJE VARTOJAMOS SĄVOKOS

Gebėjimas	Mokymosi (lavinimo ir auklėjimo) būdu išplėtotas asmens gabumas pritaikyti žinias teoriniams arba praktiniams tikslams pasiekti (Pukelis, 2009).
Įvadinis programavimo kursas (IPK)	Informatikos, programų sistemų, informacijos sistemų, informatikos inžinerijos ir kai kurių kitų krypčių studijų programose studijuojamas pirmasis studijų kursas, skirtas programavimo gebėjimams ugdytis, skirtingai vadinamas studijų programose (pvz.: „Programavimo pagrindai“, „Programavimo įvadas“, „Struktūrinis programavimas“, „Programavimas“ ir pan.).
Kompetencija	Žinių, gebėjimų ir nuostatų visuma, reikalinga tam tikrai veiklai atlikti (Europos Parlamento ir Tarybos rekomendacija dėl esminių visą gyvenimą trunkančio mokymosi kompetencijų, 2006); asmens gebėjimas naudojantis turimomis žiniomis, įgūdžiais, savybėmis ir patyrimu atlikti tam tikrą darbą. Kompetenciją lemia mokymosi ar studijų metu įgytos žinios, gebėjimai, susiformuotas požiūris ir vertinimas.
Mokymosi metodas	Veiksmų, būdų visuma mokymosi tikslui pasiekti (Jovaiša, 2007).
Mokymosi stilius	Asmens pageidaujamas būdas (pažintinis ir elgsenos) reaguoti į mokymosi užduotis, kuris kinta priklausomai nuo aplinkos ar konteksto (Peterson ir kt. 2009).
Mokymosi stilių ir mokymosi metodų dermė	Mokymosi stiliaus ypatybių ir mokymosi metodo suteikiamų galimybių tarpusavio atitiktis.
Mokėjimo mokyti kompetencija	Tai gebėjimas atkakliai ir sistemingai mokyti, planuoti, organizuoti ir reguliuoti savo mokymąsi, dalyvauti ir išsilaikyti mokymosi procese įveikiant kliūtis, efektyviai valdyti savo laiką, spręsti problemas (Europos Parlamento ir Tarybos rekomendacija dėl bendrųjų visą gyvenimą trunkančio mokymosi gebėjimų, 2006).
Programavimas	Programų kompiuteriniams įrenginiams kūrimas pagal formalias sutartines taisykles (Dagienė ir kt., 2008).
Programavimo gebėjimai	Išlavinti, išugdyti gabumai kurti įvairios paskirties programas (programuoti) (Dagienė ir kt., 2008).

Programavimo paradigma	Programuotojo apibrėžiami programos veikimo esminiai bruožai. Pavyzdžiui, objektinio programavimo paradigma reiškia, kad programos veikimas pagrįstas objektų sąveika, o funkcinio programavimo paradigma – kad programa išreiškiama funkciniais ryšiais tarp pradinių duomenų ir rezultatų be tarpinių kintamųjų. Kiekviena programavimo kalba remiasi kuria nors paradigma, o kartais ir keliomis. Pavyzdžiui, toje pačioje kalboje gali būti realizuotos struktūrinio ir procedūrinio programavimo paradigmos (Dagienė ir kt., 2008).
Programavimo mokymosi paradigma	Prielaidų, sąvokų, vertybių ir veiklų visuma, atskleidžianti programavimo gebėjimų ugdymosi strategiją.
Studijų rezultatai (siekiniai)	Teiginiai, nusakantys, ką studentas turėtų žinoti, suprasti ir gebėti padaryti, sėkmingai baigęs studijų dalyką, modulį ir (ar) visą studijų programą. Studijų rezultatai apibrėžia studento siekiamą kompetencijų lygį ir yra studijų programos aprašo dalis (Teresevičienė ir kt., 2011).
Ugdymo individualizavimas	Grupinio ugdymo forma, kai atsižvelgiama į ugdytinių individualius ypatumus ir pagal tai parenkami ugdymo turinys ir būdai (Bitinas, 2013).

DISERTACIJOJE VARTOJAMOS SANTRUMPOS

CSC2013	Tarptautinių asociacijų ACM ir IEEE parengtos rekomendacijos „Computer Science Curriculum 2013“
IK	Informatikos, informatikos inžinerijos, programų sistemų ir informacijos sistemų studijų kryptys
IKT	Informacijos ir komunikacijos technologijos
ĮPK	Įvadinis programavimo kursas
MSMMD	Mokymosi stilių ir mokymosi metodų dermė
A, M, T, P	Mokymosi stilių tipai (A – aktyvistas, M – mąstytojas, T – teoretikas, P – pragmatikas) P. Honey ir A. Mumfordo mokymosi stilių tipologijoje
In-1, In-2,...	Dėstytojai informantai
P1, P2, ..., L1, L2, ..., V1, V2, ...	Veiklos tyrimo dalyvių kodai
ILS	R. M. Felderio ir L. K. Silverman mokymosi stiliaus nustatymo klausimynas (anglų k. <i>Index of Learning Styles</i>)
LSI	D. Kolbo mokymosi stiliaus nustatymo klausimynas (anglų k. <i>Learning Style Inventory</i>)
LSQ	P. Honey ir A. Mumfordo mokymosi stiliaus nustatymo klausimynas (anglų k. <i>Learning Style Questionnaire</i>)
k	LSQ klausimyno diagnostinių kintamųjų skaičius
KR20	Kuderio-Richardsono koeficientas
n	Anketinio tyrimo imtis
N	Generalinės aibės dydis

1. MOKYMOŠI ŠTILIŲ TIPOLOGIJŲ NAUDOJIMO UGDYMOŠI PROCESĖ IR ŠTUDENTŲ PROGRAMAVIMO GEBĖJIMŲ UGDYMOŠI TEORINĖS PRIELAUDOS

1.1. MOKYMOŠI ŠTILIŲ TIPOLOGIJŲ NAUDOJIMO ASPEKTAI

Vadovaudamiesi nauja ugdymo filosofija turime orientuotis į studentų poreikius, kompetentingai jiems patarti, padėti iškilus mokymosi sunkumams, tad būtina atsižvelgti į besimokančiųjų poreikius. Pastarieji yra labai įvairūs, juos lemia skirtingos gyvenimo sąlygos, šeimoje puoselėjamos vertybės, finansinės galimybės ir pan. Po-reikiai formuoja ir besimokančiajam būdingą mokymosi, informacijos priėmimo ir apdorojimo būdą, kuris dar vadinamas mokymosi stiliumi. Įvertinti mokymosi stilių skirtumus galima bandant suprasti, kaip besimokantieji mokosi, kaip priima naują informaciją ir kaip ją apdoroja. Vis dėlto švietimo sistemoje vis dar nėra pakankamai strategijų, leidžiančių įvertinti kiekvieno besimokančiojo poreikius ir mokymosi stilių bei į juos atsižvelgti.

Pirmasis pažinimo stilių kaip asmens tipišką ar įprastą problemų sprendimo, mąstymo, supratimo ir atsiminimo būdą apibrėžė G. Allportas 1937 metais (Cassidy, 2004). Ilgą laiką mokymosi stiliais ir jų nustatymo būdais domėjosi psichologai D. Kolbas (1939), H. Witkinas (1954), I. Briggsas-Myersas (1962), A. Grasha (1974), H. Gardneris (1980), J. B. Biggsas (1987) ir kiti. Susidomėjimas mokymosi stilių teorijomis padidėjo pastaraisiais dešimtmečiais, kai atsirado individualizuotos ir personalizuotos virtualiojo mokymosi aplinkos.

Dalis mokslininkų mokymosi stilių apibrėžia kaip vidinį pažinimo procesą, vartoja tokias sąvokas kaip *apibendrintas / analitinis, impulsyvus / reflektyvus, verbalinis / neverbalinis* (Cooper, 1997, Kirby, 1988, Messick, 1976 ir kt., cituojama Coffield ir kt., 2004), kurios ir apibūdina, kaip besimokantysis koduoja, organizuoja ir apdoroja informaciją. Kiti mokslininkai, tirdami mokymosi stilius (Fleming, 2001; Chislet, Chapman, 2005), analizuoja, kaip individas pasitelkdamas pojūčius bando įsiminti informaciją. Taigi tik pagal tai vertinamas mokymosi veiksmingumas. Šiuo atveju išskiriami trys (vizualinis – regos, akustinis (garso) – klausos ir kinestetinis – lytėjimo ir veiklos) arba keturi mokymosi stilių tipai (vizualinis – regos, akustinis (garso) – klausos, skaitymo-rašymo ir kinestetinis – lytėjimo ir veiklos). Kai kurių mokymosi stilių tyrimų objektas – bendrieji suvokimo gebėjimai, kitų – tik besimokančiųjų fizinės charakteristikos. Apibrėždami mokymosi stiliaus sąvoką nemaža dalis mokslininkų

remiasi psichologine mokymosi stiliaus sąvokos prigimtimi ir teigia, kad tai – pažinimo (kognityvusis) būdas, kuris apima intelektualinę, suvokimo ir bendravimo sritis. Kitas gana dažnas mokymosi stiliaus sąvokos aiškinimas – tai geriausias arba labiau patinkantis mokymosi būdas, kurį naudoja besimokantysis. Apibrėžiant mokymosi stilių kartais akcentuojama, kad tai yra asmens savybė, kartais – kad tai yra tam tikra besimokančiojo būseną. Anot S. Cassidy, jeigu mokymosi stilius laikomas nekintamu, jis yra savybė, o jeigu mokymosi stilius laikomas kintamu, priklausomai nuo asmens patirties, – būseną (Cassidy, 2004). Kaip teigia S. Cassidy, tiek teorijoje, tiek praktikoje sąvokos *mokymosi stilius* ir *pažinimo stilius* vartojamas sinonimiškai, tačiau tai ne visai tikslu ir teisinga. Vis dėlto teigti, kad mokymosi stilius apima pažinimo stilių ir tinkamų mokymosi metodų pasirinkimą, taip pat nėra visiškai teisinga. Keletas skirtingų mokymosi stiliaus apibrėžimų pateikta 1 lentelėje.

1 lentelė

Mokymosi stiliaus apibrėžimai

Mokymosi stiliaus apibrėžimas	Apibrėžimo autorius, metai	Svarbiausias apibrėžimo akcentas
...tai pamėgtas galvojimo, informacijos apdorojimo ir supratimo būdas, taikomas pažinimo, minčių formavimo, jausmų veikimo ir vertinimo srityse.	E. Jensenas, 2001	Pažintinis aspektas
...tai biologijos ir asmens raidos apibrėžtas asmeninių savybių rinkinys, kuris, esant identiškai mokymosi situacijai, kai kuriems studentams lemia mokymosi sėkmę, kitiems – nesėkmę.	R. Dunn ir S. A. Griggs, 2000	Asmeninės savybės
...tai būdas, parodantis, kaip asmenys sutelkia dėmesį, apdoroja, perima ir įsimena naują ir sudėtingą akademinę informaciją.	R. Dunn, S. A. Griggs, J. Olsonas, M. Beasley ir B. S. Gormanas, 1995	Pažintinis aspektas
...tai pažinimo stiliaus pritaikymo mokymo(si) situacijose refleksija.	R. Ridingas ir I. Cheema, 1991	Pažintinis aspektas
...akademiniame kontekste tai gali būti fundamentali ir palyginti stabili pažinimo stiliaus ir asmenybės komponentų išraiška.	N. Entwistle, 1987	Pažinimo stiliaus ir asmeninių savybių derinys

ELSIN (anglų k. *European Learning Styles Information Network*) mokslininkų tyrėjų bendruomenės nuomone, dominuojantis mokymosi stiliaus sąvokos apibrėžimas yra šis: „Mokymosi stilius yra asmens pageidaujamas būdas (pažintinis ir elgsenos) reaguoti į mokymosi užduotis, kuris keičiasi priklausomai nuo aplinkos ar konteksto“ (Peterson ir kt., 2009). Taigi asmens mokymosi stilius yra kintanti besimokančiojo būseną.

Šiuo metu egzistuoja daug ir įvairių mokymosi stilių tipologijų bei modelių, kuriuose išskiriami skirtingi mokymosi stilių tipai. Teorijų apie mokymosi stilius įvairovė lėmė ir mokymosi stilių taksonomijų kūrimą. 1987 m. L. Curry, remdamasi mokymosi / pažinimo stilių matavimo būdais, pasiūlė sluoksninį modelį, kuriam iliustruoti vartojo svogūno metaforą. Taksonomiją sudaro keturi sluoksniai:

- 1) **Mokymo pirmenybės** (anglų k. *instructional preference*) sluoksnis nurodo individualų pageidaujamos mokymosi aplinkos pasirinkimą. Tai išorinis modelio sluoksnis, nes yra labiausiai pastebimas ir lengvai gali būti keičiamas;
- 2) **Socialinės sąveikos** (anglų k. *social interaction*) sluoksnis susijęs su socialinės komunikacijos poreikiu, atsirandančiu mokantis;
- 3) **Informacijos apdorojimo** (anglų k. *information processing*) sluoksnis nurodo individualų mąstymo būdą apdorojant informaciją;
- 4) **Asmenybės pažinimo** (anglų k. *cognitive personality*) sluoksnis – tai vidinis modelio sluoksnis, aprašomas kaip „reliatyviai pastovus asmenybės matas“, įžvelgiamas tik tuomet, kai individualus elgesys įvertinamas daugelyje skirtingų mokymosi situacijų (Riding ir Cheema, 1991, Cassidy, 2004).

2004 metais F. Coffieldas, D. Moseley, E. Hall ir K. Ecclestone atliko išsamią mokymosi stilių modelių analizę. Mokslininkų atliktos analizės ataskaitoje mokymosi stiliai skirstomi į tam tikras grupes, pasak autorių, mokymosi stilių šeimas (žr. 2 lentelę):

- 1) **Asmens charakterio** (anglų k. *constitutionally based*) šeimai priskiriamos teorijos, kurių autoriai mokymosi stilių apibrėžia remdamiesi asmens charakterio savybėmis, įtraukdami ir VAKT (anglų k. *Visual, Auditory, Kinaesthetic, Tactile*);
- 2) **Pažinimo struktūros** (anglų k. *cognitive structure*) šeimai priskiriamos teorijos, kurių autoriai mokymosi stilių apibrėžia kaip pagrindinių pažinimo struktūros bruožų, įskaitant gebėjimus, refleksiją;

F. Coffieldo mokymosi stilių teorijų taksonomija (Coffield ir kt., 2004)

Mokymosi stilių teorijų šeimos				
Asmens charakterio	Pažinimo struktūros	Stabilios asmenybės tipo	Tinkamo mokymosi būdo	Mokymosi metodų ir strategijų
Mokymosi stiliaus teorijos, jų paskelbimo metai				
Betts, 1909 Bartlett, 1932 Gordon, 1949 Sheehan, 1967 Paivio, 1971 Marks, 1973 Dunn and Dunn, 1975 Gregorc, 1977 Richardson, 1977 Torrance, 1990	Guilford, 1950 Holzman and Klein, 1954 Pettigrew, 1958 Gardner et al., 1959 Broverman, 1960 Witkin, 1962 Kagan, 1965 Hudson, 1966 Kogan, 1973 Messick, 1976 Hunt, 1978 Riding, 1991 Cooper, 1997	Myers-Briggs, 1962 Epstein and Meier, 1989 Miller, 1991 Apter, 1998 Harrison-Branson, 1998 Jackson, 2002	Kolb, 1976 Honey and Mumford, 1982 McCarthy, 1987 Kaufmann, 1989 Kirtan, 1989 Herrmann, 1995 Allinson and Hayes, 1996 Felder and Silverman, 1996 Hermanussen, Wierstra, de Jong and Thijssen, 2000	Grasha-Riechmann, 1974 McKenney and Keen, 1974 Hill, 1976 Marton and Säljö, 1976 Pask, 1976 Schmeck et al., 1977 Entwistle, 1979 Whetton and Cameron, 1984 Biggs, 1987 Weinstein, Zimmerman and Palmer, 1988 Conti and Kolody, 1990 Pintrich, Smith, Garcia and McEachie, 1991 Vermunt, 1996 Sternberg, 1998

- 3) **Stabilios asmenybės tipo** (anglų k. *stable personality type*) šeimai priskiriamos teorijos, kurių autoriai mokymosi stilių apibrėžia kaip vieną iš daugelio ir iš dalies stabilios asmenybės tipo komponentų;
- 4) **Tinkamo mokymosi būdo** (anglų k. *flexibly stable learning preferences*) šeimai priskiriamos teorijos, kurių autoriai mokymosi stilius apibrėžia kaip tinkamo mokymosi būdo pasirinkimą;
- 5) **Mokymosi metodų ir strategijų** (anglų k. *learning approaches and strategies*) šeimai priskiriamos teorijos, kurių autoriai siūlo nuo mokymosi stiliaus pereiti prie mokymosi metodų, strategijų, kryptų ir mokymosi koncepcijų (Coffield ir kt., 2004).

Atliekant disertacinę tyrimą susipažinta su keliomis dažniausiai ugdymo praktikoje naudojamomis mokymosi stilių tipologijomis (žr. 3 lentelę). Trumpai jas apžvelgsime.

3 lentelė

Ugdymo praktikoje naudojamos mokymosi stilių tipologijos

Mokymosi stiliaus teorijos / modelio pavadinimas, autorius	Iškirti mokymosi stiliaus tipai
Entwistle's Approaches and Study Skills Inventory for Students, ASSIST (Entwistle, 1979; 1998)	Įsitraukęs, organizuotas, paviršutiniškas
Kolb's Learning Styles Inventory (Kolb, 1984; 1999; 2005)	Analitikas, suvokėjas, sistemintojas, derintojas
Honey and Mumford learning styles (Honey, Mumford, 1982; 1992; 2000)	Aktyvistas, mąstytojas, teoretikas, pragmatikas
Herrmann Brain Dominance (Herrmann, 1982; 1995)	Analitinis, nuoseklus, tarpasmeninis, vaizdinis
Felder-Silvermann Learning Style Model (Felder, Silvermann, 1988; Felder, 1993)	Suvokimas: per pojūčius / intuityvus Priėmimas: vaizdinis / žodinis Organizavimas: induktyvus / deduktyvus Apdorojimas: aktyvus / refleksyvus Supratimas: nuoseklus / apibendrintas

N. J. Entwistle (2004), kurdamas mokymosi stilių teoriją, atsižvelgė į tai, kokią strategiją pasirenka besimokantieji sprenddami konkrečias mokymosi užduotis. Teorijoje išskiriami labai įsitraukusių, gerai organizuotų ir paviršutiniškai studijuojančiųjų tipai. Besimokančiųjų tikslai teorijoje siejami su vidine ir išorine besimokančiųjų motyvacija, kuri studijų metu gali kisti. Vėliau tobulinant sukurtą teorinę modelį buvo siekiama įvertinti ir sudėtingą interneto įtaką, kuri siejasi su besimokančiųjų motyvacija ir akademinė veikla (mokymo poveikis, mokymo medžiagos kūrimas, mokymosi aplinka, studijų metodai ir kt.). Tuo pat metu N. J. Entwistle modelis buvo tyrinėjamas Australijoje, Olandijoje ir JAV (Entwistle, Peterson, 2004); atlikta daugiau kaip 100 skirtingų teorinių ir empirinių tyrimų, kurių metu vertintas teorijos veiksmingumas ir daroma įtaka universitetinei pedagogikai. To rezultatas – mokymosi stilių nustatymo priemonių kaita.

D. Kolbo (1984) sukurta patirtinio mokymosi teorija ir mokymosi stilių nustatymo priemonė LSI (anglų k. *Learning Style Inventory*) sudarė pagrindą ir kitiems moksliniams tyrimams. D. Kolbas ir jo sekėjai mokymosi stilių apibrėžia kaip skirtingą mokymosi būdo pasirinkimą, kuris šiek tiek kinta priklausomai nuo mokymosi situacijų ir kuris tuo pačiu metu turi tam tikrų ilgalaikio nuoseklumo (stabilumo) požymių (Kolb ir kt.,

1999). Jis išskiria 4 etapų ciklą: 1) konkrečios patirties (anglų k. *concrete experience*); 2) reflektvyviojo stebėjimo (anglų k. *reflective observation*); 3) abstraktaus konceptualizavimo (anglų k. *abstract conceptualisation*) ir 4) aktyvaus eksperimentavimo (anglų k. *active experimentation*). Tokiu būdu išskiriami skirtingi individualūs mokymosi ir informacijos suvokimo būdai: vieni besimokantieji informaciją suvokia jausdami, remdamiesi patirtimi, tai – konkrečios patirties ir abstraktaus konceptualizavimo etapai; kiti – sistemindami ir analizuodami, o tai būtų reflektvyvaus stebėjimo ir aktyvaus eksperimentavimo etapai. Veiksmingu mokymusi D. Kolbas įvardija tokį, kai mokymosi procesas apima visus keturis etapus (Kolb, 2005). Jo nuomone, besimokančiajam būdingas vienas iš 4 mokymosi tipų: akomodacija (šiuo atveju besimokantysis prisitaiko prie aplinkos patirties ir, aktyviai eksperimentuodamas, bandydamas ir klysdamas, veikia), divergencija (konkretus patyrimas įprasminamas refleksija ir stebėjimu), konvergencija (logiškai, pragmatiškai, nešališkai priimami tikrovės iššūkiai, eksperimentuojama ir praktiškai pritaikomos naujos idėjos) ir asimiliacija (išsami analizė, tikrovė suvokiama abstrakčiai ir simboliškai). D. Kolbo teigimu (1999), asmuo, žinodamas savąjį mokymosi stilių, gali sėkmingiau dirbti komandose, spręsti kylančius konfliktus, bendrauti darbe ir namuose ar rinktis profesiją (Kolb ir kt., 1999). Todėl patirtinio mokymosi teorija ir mokymosi stilių nustatymo klausimynas yra plačiai naudojami ir išversti į daugelį kalbų (arabų, kinų, prancūzų, italų, rusų, ispanų, švedų, lietuvių ir kt.).

Mokslinius tyrimus apie tai, kaip mokosi studentai, P. Honey ir A. Mumfordas (1982) pradėjo nuo D. Kolbo sukurtos mokymosi stilių teorijos ir mokymosi stilių nustatymo priemonės LSI. 1982 m. P. Honey ir A. Mumfordas sukūrė savo mokymosi stilių nustatymo klausimyną LSQ (anglų k. *Learning Style Questionnaire*) iš 80 situacinių teiginių. Respondentai, atsakydami į klausimus, turi įvertinti savo elgseną (Honey, Mumford, 2006). Sukurtoje tipologijoje identifikuoti keturi mokymosi stilių tipai: aktyvistas, teoretikas, mąstytojas ir pragmatikas. Anot P. Honey ir A. Mumfordo, mokymasis yra keturių etapų procesas:

1 etapas – veikimas ir patirtis;

2 etapas – stebėjimas ir apmąstymas;

3 etapas – apibendrinimas ir išvadų darymas;

4 etapas – naujai įgytos kompetencijos taikymas ir naujos patirties planavimas (Honey, Mumford, 1992).

Kiekvienas mokymosi stiliaus tipas siejamas su pirmenybe konkrečiam mokymosi ciklo etapui. Autoriai pabrėžia, kad „nė vienas stilius nėra geresnis už kitą. Kiekvienas iš jų turi stipriąją ir silpnąją puses, o jų stiprybės gali būti svarbios vienoje situacijoje ir visiškai nesvarbios kitoje“ (Honey, Mumford, 1992). Mokymosi stilių nustatymo

klausimynas buvo du kartus atnaujintas – 1992 ir 2000 metais. Dabar P. Honey ir A. Mumfordo mokymosi stilių teorija plačiai naudojama verslo pasaulyje organizuojant darbuotojų mokymus ir edukologijoje, o mokymosi stilių nustatymo klausimynas išverstas į daugelį kalbų.

N. Herrmanno sukurtas visuminio mąstymo modelio pagrindas – smegenų pusrutulių ir limbinės sistemos funkcijų dominavimas apdorojant informaciją. Besimokantieji skirstomi pagal jų mąstymo (informacijos apdorojimo) pobūdį. 1982 metais mokslininkas sudarė 120 klausimų klausimyną (HBDI – anglų k. *Herrman Brain Dominance Instrument*), kuriuo remiantis nustatomas mąstymo stilius, atskleidžiantis asmens teikiamą pirmenybę emociniam, analitiniam, struktūriniam ar strateginiam mąstymui. Jo nustatytas mąstymo profilis neatspindi mąstymo kompetencijos ar įgūdžių, bet parodo asmens suteikiamą pirmenybę vienam ar kitam mąstymo būdui. N. Herrmannas teigia, kad besimokantieji turi tobulinti savo stilistinį lankstumą ir, jei reikia, plėsti savo kompetencijas (Russell, 2004). Šis modelis nors ir dažnai yra naudojamas verslo pasaulyje, pedagoginėje praktikoje taikomas labai retai.

R. M. Felderio ir L. K. Silverman (1988) sudarytą mokymosi stilių modelį sudaro penkios kategorijos: informacijos suvokimo, priėmimo, apdorojimo ir supratimo. Kiekviena iš šių kategorijų turi po dvi subkategorijas:

- Suvokimas: per pojūčius / intuityvus;
- Priėmimas: vaizdinis / žodinis;
- Organizavimas: induktyvus / deduktyvus;
- Apdorojimas: aktyvus / reflektvus;
- Supratimas: analitinis / apibendrintas.

Pasak R. M. Felderio ir L. K. Silverman (1988), asmenys silpnai priklauso kategorijai, kai balansuoja tarp abiejų kategorijos subkategorijų (t. y. abi mokymosi subkategorijos jiems yra gana priimtinos). Daugiausia sunkumų mokymosi metu patiria asmenys, labai linkę į kurią nors iš subkategorijų, kai jiems tenka mokytis priešingoje mokymosi aplinkoje (Felder, Silverman, 1988; Bloun, 2003; Felder, Spurlin, 2005).

Identifikuoti mokymosi stilių yra sudėtinga, kadangi jis yra paslėptas, latentinis kintamasis, kuris gali būti matuojamas tik netiesiogiai, nes jį lemia kiti kintamieji (Marton, 1988). Taikomi įvairūs besimokančiųjų mokymosi stiliaus nustatymo metodai: klausimynai, interviu, besimokančiųjų stebėjimas, besimokančiųjų mokymosi veiklų ir / ar atliktų darbų analizė ir pan.

Norint nustatyti mokymosi stilių bendraujant arba stebint besimokančiuosius reikia nemažai laiko, o stiliaus nustatymo tikslumas susijęs su stebėtojo gebėjimais ir patirtimi. Mokant nuotoliniu būdu mokymosi stilius nustatomas automatiškai. Šiuo

atveju virtualiojoje mokymosi aplinkoje kaupiami duomenys apie besimokančiojo atliekamas mokymosi veiklas. Juos analizuojant nustatomas besimokančiojo mokymosi stilius. Vis dėlto dažniausiai mokymosi stilius nustatomas, naudojant teorijos kūrėjo sukurtą vertinimo priemonę (pvz., klausimyną). Asmenybės įsivertinimas (anglų k. *self-report measure*), kai asmenys, atsakydami į klausimus, atskleidžia savo asmenybės bruožus, mokymosi stilius, požiūrius, interesus ir kitus su asmenybe susijusius dalykus, traktuojamas taip pat kaip ir testas (Kardelis, 2007). Tačiau, kitaip negu testuose, čia tiriamiesiems nereikia „atlikti darbo“, nes asmenybės įsivertinimo metu paprastai tiriamųjų prašoma tik atskleisti, ar jie turi tokių savybių, minčių ar jausmų, išvardytų klausimuose, ar ne. Priklausomai nuo mokymosi stiliaus sampratos, kurią apibrėžia teorijos kūrėjas, mokymosi stilių nustatymo priemonėmis vertinami skirtingi aspektai. Skiriasi ir priemonių taikymo sritys.

R. M. Felderis ir L. K. Silverman (1988) teigia, kad „besimokantieji, kurių mokymosi stilius stipriai išreikštas, sunkiai prisitaiko prie mokymo aplinkų, kuriose nėra atsižvelgiama į konkrečius atitinkamam mokymosi stiliui būdingus poreikius“ (Felder, Silverman, 1988). Tačiau tai, kad asmeniui būdingas vienas ar kitas mokymosi stilius, nereiškia, kad jis mokosi atlikdamas tik jam patinkančias mokomąsias veiklas. Nėra „teisingo“ ir „neteisingo“ mąstymo ar mokymosi stiliaus – jie visi vienodai vertingi (Felder, Brent, 2005). Kai kurie mokslininkai (Schmeck, 1988; Biggs, 2003) teigia, kad egzistuoja tiesioginis ryšys tarp mokymosi stiliaus, mokymosi būdų ir rezultatų. S. Markhamas (2004) atkreipia dėmesį, kad mokymosi stilių tyrimai neturi apsiriboti įrodymais, jog skirtingi asmenys mokosi skirtingai ir tai lemia jų mokymosi rezultatus (Markham, 2004). Apibendrinant poveikio mokymosi rezultatams tyrimus, dažniausiai laikomasi prielaidos, kad tam tikrą mokymosi stilių reprezentuojantys tiriamieji yra homogeniški ir turi tokių pačių poreikių. Todėl šiuo atveju lieka neįvertinti besimokančiųjų skirtumai.

Pedagogas, žinodamas mokymosi stilius, gali geriau pažinti besimokančiuosius. Manoma (Hennessy, 2011), kad pedagogai instinktyviai linkę mokyti tokiu būdu, kuriuo patys įprastai mokosi, tačiau toks mokymosi būdas geriausiai tinka tiems besimokantiesiems, kurių mokymosi stilius toks pat kaip ir pedagogo. Tuomet kyla klausimas, kaip sekasi mokyti tiems, kurie pratę mokyti kitokiais būdais. Pedagogas, žinantis mokymosi stilius ir susipažinęs su mažiau būdingais mąstymo ir mokymosi būdais, turi galimybę siūlyti besimokantiesiems ne tik savo paties mėgstamus mąstymo ar mokymosi būdus. Suvokęs besimokančiųjų mokymosi stilius pedagogas gali jiems padėti, sudaryti sąlygas mokyti pačiu priimtinausiu būdu. Tad besimokančiųjų mokymosi stilių teorijos gali būti taikomos mokymosi procesui individualizuoti.

Kaip jau minĖta, asmens mokymosi stilius apibrĖžia, kokioms veikloms teikiamas prioritetas. Pasak M. TereseviĉienĖs ir kt. (2004), „mokantis svarbiausia panaudoti savo talentus, galimybes ir tai, kas labiausiai mėgstama“ (TereseviĉienĖ, Oldroyd, GedvilienĖ, 2004). Besimokantysis, Źinodamas savo mokymosi stilių, geriau supras save ir tai, kaip jis mokosi. O pripaŹinęs ir išsiaiškinęs savo privalumus ir trūkumus galĖs sĖkmingiau mokytis. „Mokslinėje literatūroje mokĖjimas mokytis siejamas su besimokanĉiojo Źiniomis ir supratimu apie savo mokymąsi, asmeninį mokymosi stilių ir mokymosi strategijas, gebĖjimais nuolat vertinti ir analizuoti mokymosi patirtį, motyvuota mokymosi veikla“ (BaleviĉienĖ, PaurienĖ, 2013). Akivaizdu, kad asmeninio mokymosi stiliaus paŹinimas susijęs su mokĖjimo mokytis kompetencija, kurios svarba pabrĖžiama Europos ir Lietuvos Źvietimo dokumentuose (ETK rekomendacija dĖl bendrųjį visą gyvenimą trunkanĉio mokymosi gebĖjimų, 2006; Bendrieji Europos kalbų metmenys, 2008 ir kt.). Dokumentuose nurodoma, kad gebĖjimai ir pasirengimas išmolti mokytis turi bŹti nuolat ugdomi ir įtraukiami į visus mokomuosius dalykus, o besimokantieji asmenys turĖtų gebĖti organizuoti savo paĉių mokymąsi (Kennedy ir kt., 2009; Henard ir kt., 2012).

Daugumos mokslininkų nuomone, trŹksta įrodymų apie mokymosi stilių modelių veiksmingumą, o kai kurie mokymosi stilių modeliai net ir remiasi abejotinomis teorinėmis prielaidomis (Curry, 1990). F. J. Coffieldas (2004) mokymosi stilių kritikus skirsto į dvi stovyklas:

- pritariantys dominuojanĉioms idĖjoms (pvz., pozityvistinei metodologijai ir individualizavimo galimybei), taĉiau teigiantys, kad tam tikros konkretaus modelio ypatybĖs yra netinkamos tam tikram dalykui;
- nepritariantys dominuojanĉioms prielaidoms, kuriomis remiantis atliktas mokslinis tyrimas: sukurta teorija, suformuluotos išvados ar įrodinĖjamas poveikis mokymui (Coffield, 2004).

Tyrimai, susiję su mokymosi stilių kritika, akcentuoja keletą problemų:

- nėra konkreĉiai ir aiškiai išaiškinta *mokymosi stiliaus* sąvoka. Vis dar nesutariama, ar tai savybĖs, preferencijos, įproĉiai, strategijos ar biologiniai bruoŹai. NeŹinoma, ar mokymosi stilių prigimtis paŹintinė, neurologinĖ, psichologinė ar situacinĖ;
- nuolat ginĉijamasi ir dĖl daugelio mokymosi stilių nustatymo priemonių patikimumo ir pagrįstumo;
- nėra įtikinamų duomenų, įrodanĉių, kad taikant mokymosi stilius gerĖja mokymasis (Pashler, McDaniel, Rohrer, Bjork, 2008).

Apžvelgus įvairias sąvokos *mokymosi stilius* sampratą galima daryti išvadą, kad nėra bendro šios sąvokos aiškinimo ir apibrėžimo. Tai, kaip šią sąvoką supranta ir vartoja konkretus mokslininkas ar pedagogas, priklauso nuo pasirinktos mokymosi stiliaus tipologijos ir jos taikymo konteksto. Rekomenduojama ugdymo praktikoje taikyti mokymosi stilių tipologijas, priklausančias „tinkamo mokymosi būdo“ ir „mokymosi metodų ir strategijų“ šeimoms. Remiantis mokslinės literatūros apie mokymosi stilius analize, galima išskirti keletą mokymosi stilių taikymo sričių:

- siekiui pažinti save (savianalizei),
- mokėjimo mokytis kompetencijai tobulinti;
- mokymuisi individualizuoti ir diferencijuoti;
- profesinio mokymosi ir karjeros krypties nustatymui;
- konsultuoti (bendravimo, karjeros, mokymosi, personalo pasirinkimo, šeimos klausimais);
- grupiniams procesams organizuoti (komandoms kurti, personalo bendravimui ir bendradarbiavimui gerinti, konfliktams analizuoti ir valdyti).

Apibendrinus analizuotas mokymosi stilių tipologijas tolimesniam darbui pasirinkta P. Honey ir A. Mumfordo mokymosi stilių tipologija. Ši tipologija akcentuoja aktyvaus mokymosi veiklas, ji yra naudojama ir informatikos bei informatikos inžinerijos krypties moksliniuose tyrimuose Lietuvoje (Preidys ir kt., 2010; 2012; Kurilovas ir kt., 2014; Urbonienė, 2013; 2014). Kuriant ir tobulinant adaptyvias personalizuotas mokymosi aplinkas, kurios remiasi minėta mokymosi stilių tipologija, būtina atlikti šios tipologijos tyrimą ir edukologiniu aspektu.

Kitame skyriuje apžvelgiama programavimo samprata ir analizuojamas pradedančiųjų programavimo gebėjimų ugdymas.

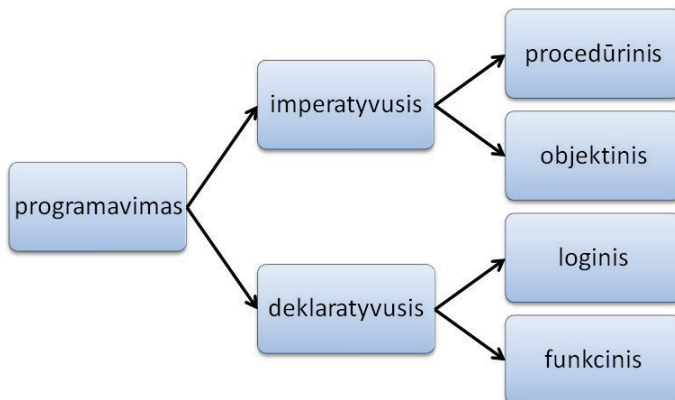
1.2. PRADEDANČIŲJŲ PROGRAMAVIMO GEBĖJIMŲ UGDYMOSI PROBLEMATIKA

Programavimo samprata. Programavimas bendrąja prasme suvokiamas kaip veiksmų, kuriuos norima atlikti, apibrėžimas (panašiai kaip planavimas). Dažnai programavimas tapatinamas su kompiuterių programų kūrimo procesu. Kartais teigiama, kad programavimas yra viena iš programų kūrimo proceso dalių, kai suplanuoti veiksmai užrašomi kuria nors programavimo kalba (tai dirbtinė kalba, skirta programoms užrašyti). Sukurtomis programomis valdomas elektroninių įren-

ginių (kompiuterių, robotų, išmaniųjų įrenginių ir kt.) veikimas. Enciklopediniame kompiuterijos terminų žodyne programavimas apibrėžiamas kaip programų kompiuteriniams įrenginiams kūrimas pagal formalias sutartines taisykles (Dagienė ir kt., 2008). Išskiriami pagrindiniai ir dažniausiai įgyvendinami programų kūrimo etapai:

1. Reikalavimų surinkimas ir analizė.
2. Įrankių (programavimo kalbos, platformos ir pan.), tinkamiausių problemai spręsti, parinkimas.
3. Programinio produkto projektavimas.
4. Programavimas – programos rašymas pasirinkta programavimo kalba.
5. Programos derinimas, klaidų taisymas.
6. Testavimas. Jei testavimo metu paaiškėja, kad nepasiektas norimas kokybės lygis, taisomos išaiškintos problemos (grįžtama į 3 žingsnį).
7. Dokumentavimas, jei reikia – vertimas į kitas kalbas.
8. Diegimas.
9. Palaikymas.

Pagal tai, kokiais principais remiantis kuriamos programos, skiriamos šios programavimo paradigmos: imperatyvusis ir deklaratyvusis programavimas. Imperatyviojo programavimo principas: programa – tai komandų, kurias reikia vykdyti, rinkinys. Komandos keičia programos būseną. Kadangi beveik visa techninė įranga dirba šiuo principu, tai didžioji dalis programų yra parašytos remiantis šia paradigma. Deklaratyviojo programavimo principas: programa aprašo, kokias sąlygas sprendinys turi tenkinti, tačiau neapibrėžia sprendinio ieškojimo algoritmo. Kiekviena realizacija gali turėti savo algoritmą (Bal, Grune, 1994; Dagienė, Grigas, 2007). Imperatyviojo ir deklaratyviojo programavimo paradigmos savo ruožtu skirstomos į atšakas (žr. 1 paveikslą).



1 pav. Programavimo paradigmų klasifikacija

Pirmosios aukšto lygio programavimo kalbos rĖmĖsi procedŭrinio programavimo paradigma. Šios paradigmos programa yra vykdomŲ instrukcijŲ sŲrašas. Programa suskaidyta į kiek įmanoma mažiau vienas nuo kito priklausomŲ moduliŲ (gali bŭti keliŲ lygiŲ: procedŭros, modulio, bibliotekos). Kiekviena tokia dalis gali turĖti tik joje matomus kintamuosius. Šią paradigmą realizuojančios kalbos: *Algol* ir iš jo išsirutuliojusios *Pascal* ir *C*. Didelis proveržis procedŭriniam programavimui buvo kitos paradigmos – objektinio programavimo – įvedimas. Tai leido duomenis ir procedŭras sujungti į vieną paketą – objektą. Objektinio programavimo paradigma reiškia, kad programos veikimas pagrįstas objektŲ sŲveika. Praktiškai kalbant, patogu vienoje vietoje turĖti ir duomenis, ir programos tekstą (t. y. „programinį kodą“), kuris dirba su tais duomenimis. Tai didelis postŭmis abstraktumo link, mat programa ima mėgdžioti realaus pasaulio objektus, programiniai elementai atrodo ir elgiasi tarsi tikrovĖs objektai. Kuriamos programos struktŭra suskirstoma į klases, kurios aprašo duomenis ir su jais susijusį funkcionalumą. Šią paradigmą realizuojančios kalbos: *C++*, *Java*. Funkcinio programavimo paradigma reiškia, kad programa išreiškia funkciniais ryšiais tarp pradinių duomenŲ ir rezultatŲ be tarpinių kintamŲjŲ, ši yra realizuota *Lisp*. O atliekant loginį programavimą sprendinio sŲlygas aprašo duomenys, su kuriais reikia dirbti, ir taisyklĖs, kaip tie duomenys yra vieni su kitais susiję. Realizuojančios kalbos: *Prolog*, *SQL*.

Norint kurti programas, neužtenka tik mokĖti programavimo kalbą ir ja programuoti, bet reikia įvaldyti visą programŲ kŭrimo procesą. TodĖl ir gebėjimas programuoti dažnai suprantamas kiek plačiau negu vien tik programos užrašymas tam tikra programavimo kalba. AIKOS¹ sistemoje apibrėžiant programuotojo kvalifikaciją nurodoma, kad asmuo, įgijęs programuotojo kvalifikaciją, gebĖs „savarankiškai arba kolektyviai atlikti sudĖtingas, asmeninės atsakomybĖs reikalaujančias projektavimo, programavimo, operacinių sistemŲ diegimo ir valdymo, duomenŲ baziŲ projektavimo, kompiuteriŲ tinklŲ projektavimo, diegimo ir administravimo, informacinių išteklŲ kŭrimo, elektroninės komercijos organizavimo, veiklos planavimo, organizavimo ir vykdymo užduotis, taikys įgytas fundamentaliąsias žinias, įvaldys vis naujas ir greitai kintančias technologijas, prisitaikys prie kintančių rinkos sŲlygŲ“. Taigi į kvalifikacijos aprašą įtraukta nemažai programuotojui bŭtinŲ dalykinių kompetencijŲ.

Šiuolaikinėje praktikoje programavimas turi kelias skirtingas taikomąsias sritis, į kurias atsižvelgia ir studijŲ programŲ kŭrĖjai. Didžiausios iš jų:

¹ AIKOS – tai atvira informavimo, konsultavimo ir orientavimo sistema, kurios pagrindinis tikslas – teikti aktualią, kokybišką informaciją apie mokymosi galimybes Lietuvoje.

- **Sisteminis programavimas** – taikomųjų programų ir programų, aptarnaujančių kompiuterio techninę įrangą, kūrimas. Tai seniausia programavimo taikymo sritis. Labai svarbu turėti objekcinio programavimo žinių. Programavimo kalbų pasirinkimas labai platus: *C++*, *C#*, *Visual Basic*, *Delphi*, *Java*. Yra sukurta daug pagalbinių programavimo priemonių (pvz., *Visual Studio*);
- **Duomenų bazių sistemų programavimas** – atskira sisteminio programavimo sritis, skirta taikomajai programinei įrangai (vadinamai duomenų bazių sistemomis), dirbančiai su dideliais duomenų kiekiais, kurti. Reikia turėti sisteminio programavimo, duomenų bazių žinių ir specifinių duomenų bazių valdymo sistemų *Oracle*, *Microsoft SQL* ar kt. žinių;
- **Programavimas internete** (kitai *web* programavimas) – įvairių tinkle veikiančių paslaugų, taip pat ir interneto svetainių kūrimas. Vartojamos programavimo kalbos: *PHP*, *Java*, *Java Script* ir kt., taip pat reikia turėti *HTML* kalbos žinių bei darbo su duomenų bazėmis (*MySQL*, *SQL*) įgūdžių;
- **Žaidimų programavimas** – įvairių kompiuterinių žaidimų kūrimas. Reikia turėti ne tik sisteminio programavimo įgūdžių, bet ir kompiuterinės grafikos, dirbtinio intelekto ir kitų sričių žinių. Programavimo kalbų spektras čia toks pat, kaip ir sisteminio programavimo. Tačiau pasirinkimas labai priklauso nuo konkretaus žaidimo idėjos;
- **Išmaniųjų įrenginių programavimas** – tai labai nauja programavimo sritis. Reikia turėti sisteminio programavimo žinių ir įgūdžių naudotis išmaniųjų įrenginių programinės įrangos kūrimo įrankiais (pvz.: *Eclipse IDE*, *Android SDK* ir kt.).

Kyla klausimas, kas svarbiau būsimajam specialistui: gerai išmanyti vieną kurią nors taikomąją sritį ar turėti visų taikomųjų sričių bendrųjų žinių; išmanyti vieną programavimo paradigmą ar turėti supratimą apie kelias? LR švietimo ir mokslo ministerijos teisės aktuose² teigiama: „Koleginės aukštojo mokslo studijos yra orientuotos į pasirengimą profesinei veiklai. Praktinis mokymas šiose studijų programose sudaro ne mažiau kaip trečdalį studijų programos. Universitetinės studijos yra labiau orientuotos į universalųjį bendrąjį išsilavinimą, teorinį pasirengimą, aukščiausio lygio profesinius gebėjimus.“ Rengiant būsimuosius programuotojus profesinei veiklai nereikėtų stengtis jų išmokyti daug programavimo kalbų, o rinktis tik kelias (geriausiu atveju dvi ar tris), labiausiai atitinkančias studijų programos pobūdį, tačiau

² LR švietimo ir mokslo ministerija: <https://www.smm.lt/mokiniams-ir-studentams/studiju-organizavimas>.

realizuojančias ne vieną programavimo paradigmą. Toliau aptarsime programavimo mokymosi pradžią pasirinkus informatikos krypties studijas.

Įvadinis programavimo kursas. Lietuvoje programavimo studijos įtrauktos į fizinių mokslų srities informatikos, informacijos sistemų, programų sistemų kryptių ir technologijos mokslų srities informatikos inžinerijos krypties studijų programas bei keletą kitų kryptių. Įvadinis programavimo kursas disertaciniame darbe vadinamas informatikos, programų sistemų, informacijos sistemų ir informatikos inžinerijos bei kai kurių kitų kryptių studijų programose studijuojamas pirmasis studijų kursas, skirtas programavimo gebėjimams ugdyti, ir studijų programose vadinamas skirtingai (pvz.: „Programavimo pagrindai“, „Programavimo įvadas“, „Struktūrinis programavimas“, „Programavimas“ ir pan.).

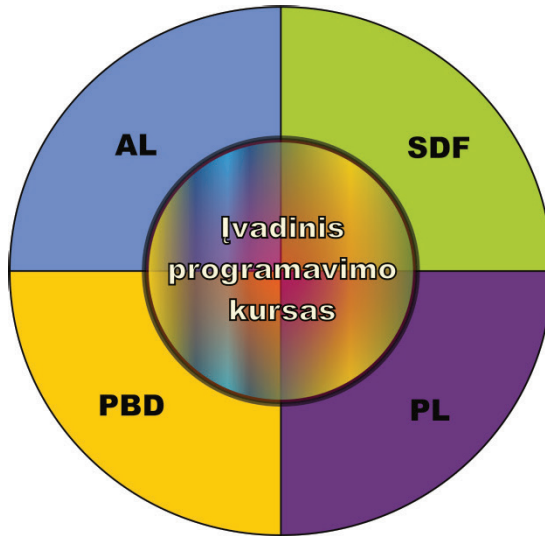
Informatikos, informacijos sistemų, programų sistemų ir informatikos inžinerijos kryptių studijų programų sandarą, kartu ir analizuojamo įvadinio programavimo kurso turinį Lietuvoje apibrėžia Informatikos studijų krypties reglamentas, tačiau didelę įtaką kuriant studijų programą turi tarptautinių asociacijų ACM ir IEEE rekomendacijos. Taip pat ir *Computer Science Curriculum 2013 (CSC2013)*, kuris yra vienas iš naujausių dokumentų, nagrinėjančių informatikos srities studijų programas. Šiose rekomendacijose išskiriama 18 informatikos mokslo žinių sričių, kurios atitinka tipines sritis informatikos studijose:

- AL – Algoritmai ir jų sudėtingumas (anglų k. *Algorithms and Complexity*);
- AR – Architektūra ir valdymas (anglų k. *Architecture and Organization*);
- CN – Skaičiavimo mokslas (anglų k. *Computational Science*);
- DS – Diskrečiosios struktūros (anglų k. *Discrete Structures*);
- GV – Kompiuterinė grafika ir vizualizavimas (anglų k. *Graphics and Visualization*);
- HCI – Žmogaus ir kompiuterio sąveika (anglų k. *Human-Computer Interaction*);
- IAS – Informacijos patikimumas ir saugumas (anglų k. *Information Assurance and Security*);
- IM – Informacijos valdymas (anglų k. *Information Management*);
- IS – Intelektinės sistemos (anglų k. *Intelligent Systems*);
- NC – Tinklai ir komunikacijos (anglų k. *Networking and Communications*);
- OS – Operacinės sistemos (anglų k. *Operating Systems*);
- PBD – Programavimo platformos (anglų k. *Platform-Based Development*);
- PD – Lygiagrečiai ir paskirstytieji skaičiavimai (anglų k. *Parallel and Distributed Computing*);

- PL – Programavimo kalbos (anglų k. *Programming Languages*);
- SDF – Programinės įrangos kūrimo pagrindai (anglų k. *Software Development Fundamentals*);
- SE – Programinės įrangos inžinerija (anglų k. *Software Engineering*);
- SF – Sistemų pagrindai (anglų k. *Systems Fundamentals*);
- SP – Socialiniai klausimai ir profesinė praktika (anglų k. *Social Issues and Professional Practice*).

Išskirtos žinių sritys nėra apibrėžiamos kaip atskiri dalykai ar moduliai, todėl sudarant įvadinio programavimo kurso turinį įtraukiama temų iš keleto susijusių sričių. Taip pat įvertinamas kitų, anksčiau studijuotų ir su programavimu susijusių, dalykų (matematikos, techninės įrangos veikimo principų, informacijos paieškos ir kt.) turinys ir rezultatai. Galimos įvadinio programavimo kurso dedamosios dalys pateiktos 2 paveiksle.

Akivaizdu, kad studijų programoje planuojamo įvadinio programavimo kurso turinys priklauso nuo studijų programos specializacijos, t. y. pasirinkimo, kurios srities programavimo žinios bus gilinamos. Ilgą laiką įvadiniai programavimo kursai buvo orientuoti į tradicines kompiuterines platformas (t. y. stacionarius arba nešiojamus kompiuterius). Vis dėlto per pastaruosius kelerius metų atsirado daug didesnė programuojamų įrenginių įvairovė, kuri taip pat akcentuojama CSC2013 rekomendacijose. Todėl rekomenduojama įvadinio programavimo kursui rinktis skirtingas platformas, pvz.: mobiliuosius įrenginius, robotus ir pan. Akivaizdu, kad įtraukus tokius dalykus į programavimo studijas bus stiprinama mokymosi motyvacija, nes tai patraukli ir iškart matoma, kaip veikia sukurti programiniai fragmentai. Taigi taikant įvairias platformas pabrėžiama programavimo sąveika su realiu išoriniu pasauliu, o tai rodo programavimo pritaikomumą. Kartu tai lemia programavimo paradigmos, komponentų bibliotekų, taikomosios programinės įrangos sąsajos (anglų k. API – *The Application Programming Interface*) ir kitų dalykų pasirinkimą. Tačiau mokslininkai įspėja, kad perėjus prie taikomojo programavimo mokymo kyla grėsmė – pasikeitus technologijoms įgytos žinios nebebus aktualios, taps bevertės, jei kartu nebus mokomasi ir technologijų veikimo pagrindų (Berry, 2013). Taigi galima ginčytis, ar tikrai verta nuo pat pradžių taip mokytis programavimo.



2 pav. Galimos įvadinio programavimo kurso dedamosios dalys pagal CSC2013 (AL – algoritmai ir jų sudėtingumas; SDF – programinės įrangos kūrimo pagrindai; PL – programavimo kalbos; PBD – programavimo platformos)

K. Bruce (2004) išskyrė tris skirtingus požiūrius, kaip ugdyti pradedančiųjų programavimo gebėjimus: 1) mokyti objektus vėliau, nedaug koreguojant objektų struktūrą; 2) pirmiausia mokyti objektus, tam pasitelkiant integruotas programavimo aplinkas *BlueJ* ar *DrJava* ir pan.; 3) nemokyti objektų, renkantis struktūrinio (procedūrinio) ar funkcinio programavimo kalbas, o objektinį požiūrį palikti papildomam kursui (Bruce, 2004). J. Bennedsenas ir M. E. Caspersenas (2004) siūlo pradėti programavimo studijas nuo objektinio programavimo sąvokų, vartojant modeliavimo kalbos *UML* elementus. Mokslininkai (Van-Roy, Haridi, 2002; Bruce, 2004; Bennedsen, Caspersen, 2004; Vujošević-Janičić, Tošić, 2008 ir kt.) svarsto probleminį klausimą, nuo ko pradėti programavimo gebėjimų ugdymą: duomenų struktūrų, algoritmų, loginio programavimo, kompiuterių techninės įrangos ar konkrečios programavimo kalbos? Tai savo ruožtu lemia įvadinio programavimo kurso turinį. Nėra vieno sprendimo ir šiuo klausimu, nors jis aktyviai svarstomas jau keletą dešimtmečių. Ne visko (programavimo metodų, programinės įrangos procesų, algoritmų, abstrakcijos, saugumo ir t. t.) gali būti mokoma nuo pirmojo studijų semestro. Mokslinėje literatūroje išskiriama keletas *programavimo mokymosi paradigmu* (Van-Roy, Haridi, 2002; Vujošević-Janičić, Tošić, 2008):

- **pradedant nuo procedūrinio programavimo** (anglų k. *imperative-first approach*) sąvokų. Procedūrinė paradigma panaši į tai, kaip veikia kompiuteris. Programa – tai vykdomų instrukcijų sąrašas ir iš esmės yra tam tikra procedūra (tarkim, skirta dviem skaičiams sudėti). Programavimas suprantamas kaip nurodymas, ką turi atlikti mašina (Austing, Barnes, Bonnette, Engel, Stokes, 1979, cituojama Factorovich, 2006);
- **pradedant nuo funkcinio programavimo** (anglų k. *functional-first approach*) sąvokų. Programavimą bandoma suvesti į matematinės funkcijas, kurių rezultatas niekada nesikeičia. Funkcija yra lyg išrinkimo veiksmas (Abelson, Sussman, 1983);
- **pradedant nuo objektinio programavimo** (anglų k. *object-oriented-first approach*) sąvokų. Didelis postūmis abstraktumo link: programos atkartoja realaus pasaulio objektus, o programiniai elementai yra realių objektų būsenos ir elgiasi tarsi realūs objektai. Vis dėlto praktiškai yra patogų – vienoje vietoje galima turėti duomenis ir kodą, kuris dirba su duomenimis;
- **pradedant nuo loginio programavimo** (anglų k. *logical-first approach*) paradigmos. Programa, parašyta loginio programavimo kalba, yra loginių reiškinių seka, išreiškianti faktus ir taisykles tam tikroje probleminėje srityje;
- **pradedant nuo techninės įrangos** (anglų k. *hardware-first approach*), t. y. nuo elektroninių grandinių, kurios gali atlikti skaičiavimus, projektavimo ir konstravimo, vėliau mokoma fizinės kompiuterio sandaros ir programavimo (dažniausiai žemo lygio programavimo kalbomis);
- **pradedant nuo algoritmų** (anglų k. *algorithms-first approach*) – pirmiausia įvedamos bazinės sąvokos, be jokios programavimo kalbos. Problemų sprendimas atliekamas žingsnis po žingsnio, dažniausiai net nesinaudojant kompiuteriu;
- **pradedant nuo sąvokų** (anglų k. *concepts-first approach*) – tiksliai ir glaustai pateikiamos visų programavimo paradigimų (imperatyvaus, loginio, funkcinio ir objektinio) sąvokos. Ši paradigma remiasi trimis pagrindiniais principais: 1) kasdienę studentų patirtį paremti informatikos mokslo principais; 2) tam tikrą laikotarpį leisti studentams dirbti su viena paradigma ir tik po tam tikro laiko pridėti antrą, dar vėliau trečią ir t. t.; 3) atskirti informatikos mokslo sąvokas nuo programavimo kalbos sintaksės;
- **platus užmojis** (anglų k. *breadth-first approach*). Tipiškas šios paradigmos įvadinis kursas apima kompiuterių programavimo pagrindus, programavimo kalbas, dirbtinį intelektą, operacines sistemas, kompiuterinę grafiką ir kt. Tai gana sudėtinga studijų pradžia, nes apimama gana daug išsamių temų.

Akivaizdu, kad įvadinio programavimo kurso turinys priklauso nuo pasirinktos programavimo mokymosi paradigmos. Pasirinkus programavimo mokymosi paradigmą kai kuriais atvejais tenka svarstyti, kokią programavimo kalbą ir programavimo aplinką pasirinkti. M. de Raadtas, R. Watsonas ir M. Tolemanas dar 2003 metais apibrėžė tokias programavimo kalbos pasirinkimo priežastis: 1) paklausa darbo rinkoje; 2) pedagoginė kalbos nauda; 3) švietimo politika; 4) objektinio programavimo paradigma; 5) grafinė vartotojo sąsaja; 6) programinės įrangos kaštai; 7) mokomosios medžiagos kiekis; 8) turimos techninės įrangos galimybės. Profesinėse studijose dažniausiai pasirinkimą lemia pirmosios dvi priežastys, nors visiškai atmesti negalima ir likusiųjų. V. Dagienė ir J. Urbonienė (2010) atliko programavimo kalbų tinkamumo pradedančiųjų mokymuisi vertinimą pagal 17 kriterijų, suskirstytų į 4 grupes, apimančias tam tikras sritis, – mokymąsi, kūrimą ir aplinką, pagalbą ir prieinamumą bei taikymą. Nustatyta, kad daugiausia kriterijų atitinka ir mokytis tinkamiausios būtų *C++*, *Eiffel*, *Java* ir *Python* programavimo kalbos (Dagienė, Urbonienė, 2010).

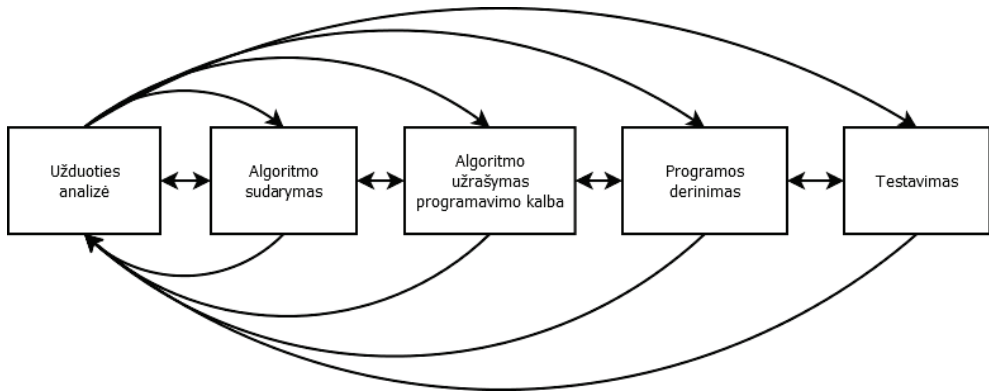
Apibendrinant galima teigti, kad įvadinio programavimo kurso turinį lemia studijų programos kryptis (specializacija), o esant didelei programavimo mokymosi paradigmų ir kalbų įvairovei galima organizuoti pradedančiųjų programavimo gebėjimų ugdymą su skirtingomis programavimo kalbomis arba įvesti sąvokas nesiremiant jokia konkrečia programavimo kalba.

Pradedančiųjų programavimo mokymo moksliniai tyrimai dažniausiai apima programų kūrimo procesą ir supratimą, mąstymo modelius, mokymosi būdus, būtinas programavimo žinias ir gebėjimus (Robins ir kt., 2003). Mokslinio redukcionizmo principo esmė: sudėtinga sistema gali būti paaiškinta, išskaidyta (redukuota) į sudėtingesnes dalis. Programavimo gebėjimų ugdymą analizavo T. J. McGill ir S. E. Volet, remdamosi P. Bayman ir R. E. Mayerio modeliu (1988). Pabrėžiama, kad besimokantieji turi įgyti trijų tarpusavyje susijusių tipų žinias:

1. **Sintaksės.** Tai programavimo kalbos konstrukcijos ir jos vartojimo taisyklės.
2. **Sąvokų.** Tai programavimo sąvokos ir principai, kurių reikia besimokančiajam, kad sudarytų programų veikimo modelius.
3. **Strategijos.** Tikslingai naudojantis sintaksės ir sąvokų žiniomis, įgyvendinami programavimo užduočių sprendimai. Strategijos žinios apima ir problemų sprendimo įgūdžius, o tai jau programavimo specifika (McGill, Volet, 1997).

V. Dagienė ir J. Urbonienė programavimo mokymą skirsto į du etapus: 1) programavimo kalbos sintaksės, semantikos išmokymas; 2) gebėjimas taikyti programavimo techniką ir metodus (Dagienė, Urbonienė, 2010).

Šiuolaikinių programų kūrimo procesą sudaro keletas dalių. Suprantama, kad mokantis programuoti nepradedama iškart kurti sudėtingų programų, tačiau ir paprasta programavimo užduotis sprendžiama remiantis programų kūrimo etapais, dažniausiai praleidžiant programinių priemonių pasirinkimo, programų dokumentavimo, palaikymo ir diegimo etapus. Pradedančiųjų programavimo proceso (programavimo užduoties sprendimo) struktūra pavaizduota 3 paveiksle.



3 pav. Programavimo proceso (programavimo užduoties sprendimo) etapai

Pirmasis programavimo proceso etapas – užduoties analizė. Šiame etape programuotojas turi išsiaiškinti būsimosios programos reikalavimus, numatyti pradinis duomenis ir tai, kokie bus gaunami programos veikimo rezultatai. Pastebėta, kad jau užduoties analizės etape nemažai daliai pradedančiųjų iškyla sunkumų, todėl reikia skirti dėmesio užduočių analizei: skaityti, suprasti, apibrėžti pradinis duomenis ir rezultatus.

Antrasis programavimo proceso etapas – tai veiksmų plano, kaip iš pradinių duomenų gauti rezultatus, sudarymas. Šis etapas vadinamas užduoties realizavimo algoritmo kūrimu. Priklausomai nuo užduoties sudėtingumo, algoritmas gali būti sudaromas parenkant ir pritaikant tipinius algoritmus naujos užduoties kontekste arba kuriamas tuo atveju, kai užduotis nėra tipiška ir reikalauja kūrybiškumo. Šiame etape reikia suprasti programavimo konceptus ir gebėti taikyti matematinės logikos žinias, skaidyti didelę užduotį į sudėtines dalis.

Trečiajame etape užduoties algoritmas užrašomas kuria nors konkrečia programavimo kalba. Šiame etape reikia parinkti duomenims ir rezultatams tinkamas duomenų struktūras, išmanyti programavimo kalbos sintaksę ir semantiką bei gebėti pritaikyti šias žinias sprendžiant realią problemą. Programavimo kalbos sintaksė – tai taisyk-

lių visuma, apibrėžianti konstrukcijų (sakinių) struktūrą. Sintaksė nusako, kaip turi būti užrašytos programavimo kalbos konstrukcijos, bet nenusako jų prasmės. O taisyklių visuma, apibrėžianti konstrukcijų (sakinių) prasmę ir jų santykius, vadinama semantika (Dagienė, Grigas, Jevsikova, 2008). „Mokantis programavimo technikos, kalba yra tik priemonė bendroms programavimo sąvokoms išreikšti ir taikyti. Pasi-taiko klaidinga programavimo mokymosi samprata: manoma, jog tai yra mokymasis užrašyti uždavinio sprendimą programos tekstu panaudojant programavimo kalbos konstrukcijas. Programos rašymas yra tik vienas iš programavimo įgūdžių“ (Dagienė, Urbonienė, 2010). Tačiau šiuolaikiniai besimokantieji vis rečiau imasi kurti algoritmą ar rašyti programą „nuo nulio“. Dažniau bandoma pasirinkti sprendimo būdą, rasti jį internete ar sudaryti veikiančią programą, sudėliojant ją iš turimų pavyzdžių fragmentų ir juos šiek tiek keičiant. Tokiu būdu nekuriamas „savo“ užduoties sprendimo, o bandoma adaptuoti surastą, kurio dažniausiai besimokantysis net nesupranta, nes užduoties sprendimas yra aukštesnio lygmens programinė realizacija (t. y. sprendžiant užduotį panaudojama tai, kas pradedančiajam dar nėra žinoma). Remiantis turima dėstyto patirtimi, galima teigti, kad tokį programavimo būdą renkasi dauguma pradedančiųjų. Tokiu būdu labai svarbu gebėti rasti tinkamus programinius fragmentus, reikia skirti dėmesio informacijai ieškoti, atrinkti, apibendrinti ir įvertinti.

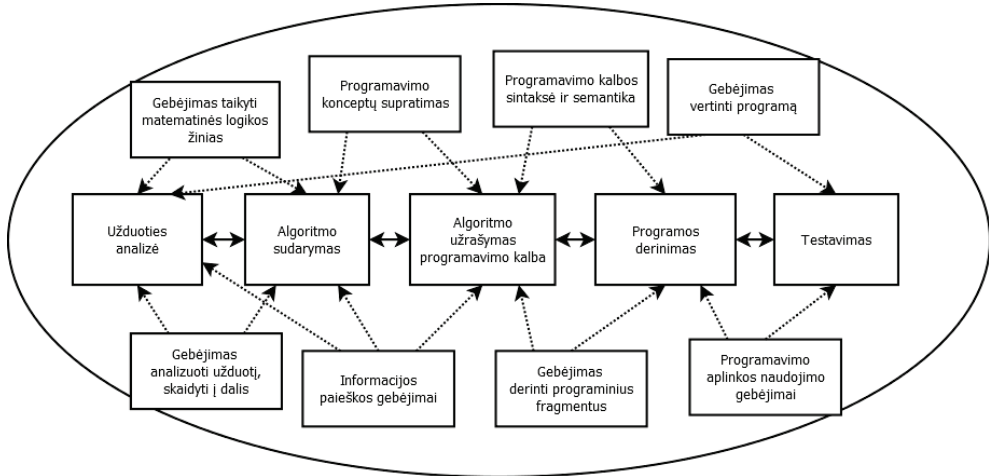
Komponuojant galutinį ir teisingą užduoties sprendimą iš atskirų dalių arba užrašius savo sukurtą algoritmą, pereinama į programos derinimo etapą. Derinant programą reikia žinoti programavimo kalbos sintaksę bei suprasti kompiliatoriaus pateikiamus pranešimus apie aptiktas klaidas. Šiame etape naudojantis programavimo aplinkos galimybėmis galima greičiau išsiaiškinti padarytas klaidas. Vadinasi, svarbu ugdytis gebėjimus naudotis programavimo aplinka.

Testavimo metu įvertinama, ar parašyta programa teisinga. Parinkus testinius duomenų rinkinius, siekiama įsitikinti, ar visais atvejais sukurtoji programa veikia taip, kaip buvo tikimasi. Tinkamas testinių duomenų parinkimas tiesiogiai susijęs su tinkamai atlikta užduoties analize.

Apibendrinami galime teigti, kad norint išspręsti ir paprastą programavimo užduotį būtina turėti daug įvairių žinių ir gebėjimų. Galima teigti, kad sklandžią pradedančiųjų programavimo proceso eigą lemia šie gebėjimai (žr. 4 paveikslą):

- gebėjimas analizuoti užduotį;
- gebėjimas taikyti matematinės logikos žinias naujame kontekste;
- programavimo konceptų supratimas;
- programavimo kalbos sintaksės ir semantikos žinios bei gebėjimas jas taikyti;
- informacijos paieškos gebėjimai;

- programiniŲ fragmentŲ derinimo gebĖjimai;
- sukurtoš programos vertinimo gebĖjimai;
- programavimo aplinkos naudojimo gebĖjimai.



4 pav. Pradedančiųjų programavimo gebĖjimai, lemiantys sklandžią programavimo proceso eigą

Sudarant įvadinio programavimo kursą būtina skirti dėmesio kiekvienai išskirtai kategorijai, parinkti atitinkamas programavimo uŲduotis, planuoti mokymosi veiklas ir parinkti tinkamus mokymosi metodus. MatematinĖs logikos žinios dažniausiai įgyjamos studijuojant matematiką, o studijuojant programavimą jos taikomos naujame kontekste. Aprašyti pradedančiųjų programavimo gebĖjimai atskleidžia ir įvadinio programavimo mokymosi tikslus, t. y. formuoti programavimo sampratą, įgyti pasirinktos programavimo kalbos sintaksĖs žinių ir ugdytis gebĖjimus analizuoti uŲduotį, ieškoti teisingo sprendimo, jį įgyvendinti, testuoti ir įvertinti gautus rezultatus.

Pradedančiųjų programavimo gebĖjimų ugdymosi sėkmės veiksniai. Aukštojoje mokykloje įvadinis programavimo kursas dažniausiai studijuojamas pirmaisiais studijų metais. Dauguma kurso klausytojų – 18–22 metų jaunuoliai. Vertinant biologiniu (fizinĖs kaitos) aspektu, tai – suaugusieji. Tačiau tapsmas suaugusiuoju yra nuoseklus procesas – pereinama iš vaikystĖs per paauglystę į suaugusiųjų gyvenimą. Vis dėlto nėra griežtŲ ribŲ, nurodančių, kada tai įvyksta (Teresevičienė ir kt., 2006). Šiuo metu profesines (taip pat ir programavimo) studijas vis daugiau renkasi ne tik jauni, bet ir vyresni žmonės, turintys įvairios asmeninės ir profesinės patirties. Esminiu

skirtumu tarp pedagoginių ir andragoginių idėjų taikymo laikomas besimokančiojo savarankiškumas.

Dabartinė studentų karta (vadinama *millenium*, arba *digital natives*, anot M. Prensky, 2013) užaugo technologijomis pripildytoje aplinkoje. Sparti IKT kaita ir teikiamos galimybės sudaro sąlygas nuolat palaikyti ryšį, bendrauti su šeimos nariais, giminaičiais, draugais, bendramoksliais ir kt. Anot J. L. Frando (2006), naujosios kartos atstovams dažniausiai nepriimtina ir nesuvokiama izoliacija nuo kitų žmonių ar atsiribojimas nuo technologijų (Frando, 2006). Svarbu į tai atkreipti dėmesį organizuojant šiuolaikinių studentų mokymąsi. Mokymo ir mokymosi procesai gali būti įvardyti kaip aplink supančio pasaulio kūrimas, atrandant galimybes, mokantis iš savo patirties bei kaupiant patirtį: tai svarbu ir studentams, ir dėstytojams – patiriama mokymosi ir darbo prasmė (Bubnys, Žydžiūnaitė, 2012). Apibūdinant mokymosi ypatumus aukštojoje mokykloje, pripažįstamas mokymosi kompleksiskumas, kai individuali studento patirtis, kurso ir mokymosi kontekstas (kurso turinys, forma, mokymosi metodai, vertinimas) siejamas su technologijų apsupta mokymosi aplinka, mokymosi rezultatais ir studento požiūriu į kontekstą bei mokymąsi (Zuzevičiūtė, Teresevičienė, 2007). Taigi mokant šiuolaikinius studentus būtina parinkti ir taikyti tinkamas ar net atrasti naujas andragogines strategijas.

J. Lowe (2000) išskiria pagrindinius besimokančiojo motyvus: profesinius, asmenybės tobulėjimo ir socialinių ryšių. Pasak autoriaus, profesiniai motyvai yra svarbūs nuo 18 iki 40 metų, kai žmonės pradeda dirbti, įgyja socialinį statusą, kelia kvalifikaciją, tampa geresniais specialistais, daro karjerą. Asmenybės tobulėjimo motyvai yra svarbesni vyresniems, kai tam jie turi daugiau laiko, finansinių galimybių. O skatinami socialinių ryšių motyvo žmonės ieško ryšių su kitais žmonėmis, bendrauja, dalyvauja įvairiose diskusijose (Lowe, 2000). Bendravimą ir bendradarbiavimą su bendramoksliais, ugdytojų ir kitų asmenų pagalbą besimokančiajam bei grupės mokymosi atmosferą kaip svarbiausius veiksnius akcentuoja ir V. Lukošūnienė (2014), tyrusi suaugusiųjų mokėjimo mokytis kompetencijų ugdymąsi ir raišką. Analizuojant ugdymosi sėkmę, mokymosi motyvacijos klausimas yra laikomas vienu iš svarbesnių, jei ne pačiu svarbiausiu dalyvaujantiems mokymo ir mokymosi procese – tiek besimokantiems, tiek organizuojantiems mokymą (Juozaitis, 2003). Visa tai, ko žmogus siekia, nulemia asmenybės kryptingumą. Motyvai nurodo, kas skatina suaugusįjį veikti. Jie yra suvokiami kaip poreikiai, interesai, polinkiai, įsitikinimai, idealai, aktyvumas, nuostatos. Kadangi žmogaus aktyvumo pagrindas yra jo poreikiai, motyvacija yra neatsiejama nuo tikslo ir poreikio. Ji yra svarbi ne vien dėl to, kad nulemia, – tai gali paskatinti susiformuoti tvirtesnę elgesį,

bet reikalinga ir tam, kad kiekvieno suaugusiojo elgesys būtų orientuotas į tikslą (Kuncaitis, 2009).

Programavimas yra sudėtinga intelektinė veikla ir jau beveik 40 metų visuotinai pripažįstama, kad programavimo mokymosi pradžia yra itin sunki. A. McGettricko ir kt. (2005) nuomone, tai – vienas iš septynių didžiųjų iššūkių siekiantiems informatico išsilavinimo (McGettrick ir kt., 2005; Bennedsen, Caspersen, 2008). Suprantama, kad sudėtinga veikla iš besimokančiojo reikalauja tam tikrų asmeninių būdo bruožų: kantrybės, užsispyrimo, kruopštumo, atidumo detalėms ir pan. Mokslinėje literatūroje pateikiama programavimo mokymosi sunkumus lemiančių priežasčių klasifikacija: 1) ugdytojo taikomi mokymo metodai; 2) besimokančiųjų pasirenkami mokymosi metodai; 3) besimokančiųjų gebėjimai ir nuostatos; 4) programavimo prigimtis; 5) psichologiniai motyvai (Gomes, Mendes, 2007; Urbonienė, 2014). Remiantis šia pradedančiųjų programavimo mokymosi sunkumus lemiančių priežasčių klasifikacija, galima išžvelgti atitinkamus veiksnius, kurie veikia pradedančiųjų programavimo gebėjimų ugdymą. Tad disertaciniame darbe išsamiau tyrinėjami pradedančiųjų programavimo gebėjimų ugdymosi proceso veiksniai į studentus orientuoto ugdymo kontekste.

1.3. PRADEDANČIŲJŲ PROGRAMAVIMO GEBĖJIMAMS UGDYTIŠ TAIKOMI MOKYMOŠI METODAI

Pastaraisiais metais įvyko pokytis – bendra aukštųjų mokyklų didaktikos (re) orientacija į mokymą, o ne mokymą, į probleminius, projektinius, komandinius studijų metodus, atviras, savarankiškas, taip pat ir studijas nuotoliniu būdu, o ne į tradicinę frontalinę medžiagos dėstymą paskaitose ir privalomos literatūros skaitymą (Glosienė, 2006). Suaugusiųjų mokymosi kontekste mokymosi metodai apibrėžiami kaip pasikartojančių veiksmų visuma, tinkanti įvairiems dalykams mokytis ir išmolti (Teresevičienė, Gedvilienė, Zuzevičiūtė, 2006).

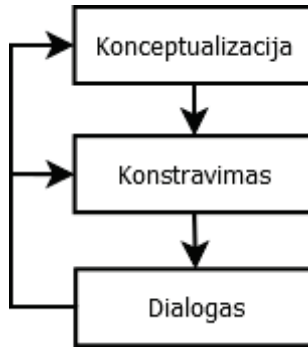
Mokymosi metodų yra labai daug ir įvairių, skirtingos ir jų klasifikacijos. L. Jovaiša (2007) metodus skirsto į:

- informacinius, kurie taikomi siekiant perteikti informaciją (metodų pavyzdžiai: demonstravimas, įvairių šaltinių nagrinėjimas ir pan.);
- praktinius-operacinius, skirtus tam tikros veiklos atlikimo mokėjimams išsiugdyti (metodų pavyzdžiai: pratybos, praktikos, laboratoriniai darbai ir pan.);

- kűrybinius, skatinančius studento savarankiškumą, kritinį ir kűrybinį mąstymą (metodų pavyzdžiai: į problemų sprendimą orientuotas studijavimas, atvejo analizė ir pan.).

Kiti autoriai (Biggs, Tang, 2007; Lewis, Reinders, 2008; CEDEFOP, 2008) išskiria pasyviuosius ir aktyviuosius studijų metodus, priskirdami juos skirtingoms studijų paradigmoms – tradicinei ir šiuolaikinei, orientuotai į aktyvias, savarankiškas studentų studijas. Akademinėje literatűroje (Rajeckas, 1999; Jovaiša, 2002; Teresevičienė, Gedvilienė, 1999, 2003; Teresevičienė, Gedvilienė, Zuzevičiūtė, 2006; Petty, 2007, 2008; Lewis, Reinders, 2008) pristatoma gausi mokymosi metodų įvairovė. Įprasta, kad planuodamos studento darbo krűvį aukštosios mokyklos paskirsto auditorinio darbo valandas paskaitoms, seminarams, pratyboms, laboratoriniams darbams, konsultacijoms ir kt. Kaip teigia M. Teresevičienė, „paskaitos, seminarai, praktiniai užsiėmimai, laboratoriniai darbai yra makrolygmens“ (Teresevičienė ir kt., 2011) studijų metodai. Tačiau makrolygmens studijų metodai neatskleidžia visos kurso studijų metu atliekamos dėstytojo ir studentų veiklos, tad toks studijų metodų skirstymas pasiteisina tik studijų organizavimo aspektu. „Mikrolygmenyje veikiančios didaktinės prielaidos, tai yra studijų metodai, kuriuos galima pasitelkti ir paskaitoms, ir seminarams, ir laboratoriniams darbams“ (Teresevičienė ir kt., 2011). Natűralu, kad ir Lietuvoje, ir užsienyje įvadinio programavimo studijoms taikomi įvairūs mokymo ir mokymosi metodai, kadangi dėstomos įvairios programavimo kalbos, naudojamos skirtingos programavimo mokymosi paradigmos ir strategijos (Van-Roy, Haridi, 2004; Vitkutė-Adžgauskienė, Vidžiūnas, 2012). Kaip jau minėta anksčiau, naudojamos ir virtualiosios mokymosi aplinkos (pvz., Moodle) priemonės, ir įvairios programavimui mokyti skirtos interaktyviosios aplinkos (pvz., Scratch, Alice, code.org, programmr.com ir kt.). Dėl sudėtingumo programavimas nėra patrauklus, todėl norint juo sudominti besimokančiuosius reikia pateikti jį kuo paprasčiau, aiškiau ir patraukliau, nes „viena iš svarbiausių dėstytojams tenkančių užduočių yra organizuoti studijų procesą taip, kad jis atitektų individualius studentų tobulėjimo poreikius bei skatintų studentus prisiimti atsakomybę už savo studijavimą“ (Heron, 1999). Tam, kad dėstytojas galėtų rekomenduoti tinkamus studijų metodus, jis turi įvertinti konkrečių metodų privalumus ir trūkumus siekdamas numatytų studijų rezultatų. Dauguma studijų metodų taikomi įvairių kursų studijoms labai panašiai, tačiau kai kurie iš jų labiau atliepia dalyko poreikius ir taikomi gan specifiskai mokyti įvadiniam programavimui. Šių metodų aprašai pateikti 9 priede.

J. T. Mayesas ir C. J. Fowleras (1999) aprašė mokymosi ciklą su grįžtamuju ryšiu, kurį sudaro trys pakopos: konceptualizacija, konstravimas ir dialogas (žr. 5 paveikslą).



5 pav. J. T. Mayeso ir C. J. Fowlero (1999) mokymosi ciklas

Mokymosi ciklas – tai nuoseklus procesas su grįžtamuju ryšiu – laipsniško supratimo tobulinimo ciklas. Atitinkamai mokymasis pereina tris pakopas: nuo konceptualizacijos į konstravimą ir dialogą. Konceptualizacijos pakopa – tai sąveika tarp besimokančiojo iš anksto egzistuojančios žinių sistemos ir pedagogo žinių. Konstravimo pakopa – sąvokų kūrimas ir derinimas, jų vartojimas sprendžiant prasmingus uždavinius. Dialogo pakopoje besimokantieji ir pedagogas bendradarbiaudami tikrina ir kuria naujas sąvokas. Šios trys mokymosi ciklo pakopos yra glaudžiai susijusios su mokymosi teorijomis: konceptualizacija – su kognityvine mokymosi teorija, nes remiasi sąvokomis ir jų ryšiais; konstravimas – su konstruktyvistine mokymosi teorija, nes jo tikslas – įgyti naujų žinių aktyvia veikla; dialogas – su socialinio konstruktyvizmo mokymosi teorija, nes šioje pakopoje akcentuojamas bendravimas ir bendradarbiavimas. „Pedagoginėje psichologinėje literatūroje nurodomi mokymosi teorijų fundamentalūs filosofiniai skirtumai, tačiau mokymo projektuotojai linkę tikėti, kad tai, kas veikia mokyme, tai teorijų mišinys, subtili mokymosi teorijų kombinacija“ (Mayes, Fowler, 1999). Toliau išsamiai aptarsime pradedančiųjų programavimo gebėjimų ugdymosi atvejį, kiekvienam J. T. Mayeso ir C. J. Fowlero ciklo etapui numatydami tinkamus mokymosi metodus ir būdus.

Konceptualizacija. Šiame etape aiškinama nauja mokymo medžiaga, ji siejama su turima besimokančiųjų patirtimi. Naujos sąvokos įvedamos tokiais būdais:

1. Paskaitų metu pateikiamos naujos sąvokos ir apibrėžimai. Šiuo atveju dažniausiai taikomi informaciniai mokymo metodai: paskaita, demonstracija,

algoritmų ir programų vizualizavimas, pavyzdžių analizė – programos teksto aiškinimas.

2. Besimokantieji patys savarankiškai išsiaiškina sąvokų reikšmę iš jiems pateiktos (pvz., virtualiojoje mokymosi aplinkoje) mokomosios medžiagos turinio. Taikomi mokymosi metodai: lektūra, vaizdo pamokos, minčių lietus, koncepcijų žemėlapis, algoritmų ir programų vizualizavimas, diskusijos, programos teksto skaitymas ir aiškinimasis. Gali būti atliekamos įvairios užduotys ir pratimai, padedantys suvokti sąvokas, išsiaiškinti pritaikymo sritis.

Konstravimas. Šiame etape įtvirtinamos naujos sąvokos, nes yra vartojamos sprendžiant realias problemas. Atliekami kūrybiniai praktiniai darbai (pratimai) – rašomos programos individualiai, grupėmis arba porose, braižomos algoritmų blokinės schemas, naudojamos problemų sprendimo strategijos.

Dialogas. Šiame etape žinios įtvirtinamos dalijantis įgyta patirtimi, tikrinant ir vertinant žinias. Taikomi metodai: darbas grupėse, diskusijos; vertinamas savo ir grupės narių darbas, atliekami savikontrolės testai, individualūs praktiniai darbai.

Ankstesniame skyriuje aprašyta pradedančiųjų programavimo gebėjimų ugdymosi struktūra (žr. 4 paveikslą) gali būti siejama su J. T. Mayeso ir C. J. Fowlero mokymosi ciklu. Su konceptualizacijos etapu glaudžiai susiję gebėjimai taikyti matematinės logikos žinias naujame kontekste, suprasti programavimo konceptus, įgyti programavimo kalbos sintaksės ir semantikos žinių. Su konstravimo etapu – gebėjimai analizuoti užduotį, ieškoti informacijos, derinti programinius fragmentus, naudotis programavimo aplinka, vertinti sukurtą programą. Su dialogo etapu – gebėjimas užtikrinti grįžtamąjį ryšį, vertinti bei įsivertinti žinias ir gebėjimus. Remiantis aprašytais J. T. Mayeso ir C. J. Fowlero mokymosi ciklo etapais galima suklasifikuoti mokymosi metodus, taikomus ugdant programavimo gebėjimus, ir virtualiosios mokymosi aplinkos priemones (žr. 4 lentelę).

Mokymosi ciklo etapų ir mokymosi metodų sąsajos

J. T. Mayeso ir C. J. Fowlero mokymosi ciklo etapai (1999)	Mokymosi metodai, būdai	Virtualiosios mokymosi aplinkos priemonės
Konceptualizacija	Įtraukianti paskaita, demonstravimas, algoritmų ir programų vizualizavimas, programos teksto skaitymas (pavyzdžių analizė) ir veikimo sekimas, lektūra, vaizdo pamokos, minčių lietus, koncepcijų žemėlapis, diskusija	Mokomoji medžiaga pateikta įvairiais formatais: <i>pages, file, url, glossary</i> ; minčių žemėlapiu įrankis: <i>mindmap</i> ; bendravimo priemonės: <i>forum, chat, messaging</i>
Konstravimas	Kūrybiniai praktiniai darbai (pratimai) – programos rašomos individualiai, grupėmis arba porose, vizualizuojami algoritmai ir programos, naudojamos problemų sprendimo strategijos	Praktinės užduotys: <i>assignment, wiki, blog, lesson, mindmap, workshop, VPL</i> ; bendravimo priemonės: <i>forum, chat, messaging</i>
Dialogas	Darbo grupėse metodai, diskusijos, vertinamas savo ir grupės narių darbas, atliekami savikontrolės testai, individualūs praktiniai darbai	Vertinimo priemonės: <i>quiz, feedback</i> ; bendravimo priemonės: <i>forum, chat, messaging</i>

Planuojant pradedančiųjų programavimo gebėjimų vertinimą ir įšivertinimą pasitelkiama Bloomo taksonomija, kuri plačiai naudojama aprašant įgytas žinias ir gebėjimus, renkant mokymosi ir vertinimo metodus. Bloomo taksonomijoje išskirtos trys mokymosi proceso sritys: pažinimo, emocinė ir psichomotorinė. Rengiant mokymo programas ir planuojant besimokančiųjų vertinimą svarbiausia laikoma pažinimo sritis. Remiantis B. S. Bloomu (1956), pažinimo srities tikslai skirstomi į šešis lygius, pradedant nuo paprastesniųjų ir pereinant prie sudėtingesniųjų:

- **žinojimas** – tai informacijos atsiminimas, atgaminimas, atpažinimas;
- **supratimas** – gebėjimas informaciją suvokti, išžvelgti jos prasmę, susieti su kita medžiaga;
- **taikymas** – gebėjimas panaudoti idėjas, taisykles, metodus konkrečiose situacijose;
- **analizė** – mokėjimas išžvelgti informacijos sandarą ir išdėstymą, suskaidyti visumą į dalis;
- **sintezė** – pavienių elementų ar dalių derinimas, jungimas, naujų elementų kūrimas;
- **vertinimas** – kokio nors dalyko vertingumo nustatymas, apibūdinimas pagal tam tikrus kriterijus (Bloom ir kt., 1956; 1964).

Su aukštesniaisiais mąstymo įgūdžiais siejama analizė, sintezė ir vertinimas. Tam, kad būtų galima kurti kažką nauja, reikia turėti žinių, jas suprasti, mokėti pritaikyti, išskirti pavienius elementus, juos naujai konstruoti ir gebėti įvertinti savo kūrybos rezultatus.

XX a. dešimtajame dešimtmetyje Bloomo taksonomija buvo peržiūreta ir pakoreguota. Taksonomiją atnaujino B. S. Bloomo mokinys L. Andersonas kartu su kolega D. Krathwohlu. Esminis šių dviejų taksonomijų skirtumas – pakoreguotame variante aukščiausiam lygmeniui priskiriamas ne vertinimas, o kūryba. Atnaujintoje taksonomijoje pakeisti pažinimo lygmenų pavadinimai ir tvarka: įsiminimas, supratimas, taikymas, analizė, vertinimas ir kūryba. L. Andersonas ir D. Krathwohlas peržiūrėjo ir pakoregavo kiekvieno taksonomijos lygmens aktyviusius veiksmožodžius, susijusius su besimokančiųjų atliekama mokymosi veikla (žr. 5 lentelę). Numatytieji veiksmožodžiai apibūdina daug ir įvairių mokymosi veiklų, veiksmų, procesų, kurie vyksta klasėje. Vis dėlto šių veiklų sąrašė nebuvo numatyta informacijos ir komunikacijos technologijomis (IKT) grįstų veiklų, aktualių šiuolaikinio ugdymo procese. Todėl 2008 metais A. Churchesas papildė L. Andersono ir D. Krathwohlo atnaujintą taksonomiją, įtraukdamas į ją IKT grįstas mokymosi veiklas. Taksonomija pavadinta „skaitmenine Bloomo taksonomija“ (anglų k. *Bloom’s Digital Taxonomy*) (žr. 5 lentelę).

5 lentelė

Atnaujintos Bloomo taksonomijos lygmenys ir juos atitinkančios mokymosi veiklos

Taksonomijos lygmenys	Besimokančiųjų atliekami veiksmai (Anderson ir Krathwohl, 2001)	Mokymosi veiklos, naudojant IKT (Churches, 2009)
Įsiminimas	Atpažinti, išvardyti, aprašyti, nustatyti, pakartoti, surasti, įvardyti	Sudaryti sąrašus, išskirti, pabrėžti pagrindines sąvokas ar frazes (anglų k. <i>highlighting</i>), kaupti nuorodas (anglų k. <i>bookmarking</i>) ir jas tvarkyti, naudoti socialinius tinklus, ieškoti informacijos (anglų k. <i>googling</i>)
Supratimas	Interpretuoti, apibendrinti, perfrazuoti, klasifikuoti, palyginti, paaiškinti, iliustruoti	Naudoti išplėstinę paiešką, atlikti loginę paiešką, rašyti tinklaraštį, žymėti (anglų k. <i>tagging</i>), klasifikuoti failus ir interneto išteklius, komentuoti, prenumeruoti (anglų k. <i>subscribing</i>), rengti ir bendrinti dokumentus bendradarbiaujant su kitais, rašyti vikį, sudaryti minčių žemėlapi
Taikymas	Įgyvendinti, atlikti, vykdyti, naudoti	Manipuliuojant įranga ir programomis pasiekti išskeltus tikslus ar gauti reikiamus rezultatus, žaisti (kompiuterinius žaidimus), įkelti ir dalytis turima informacija (anglų k. <i>uploading and sharing</i>), redaguoti interneto medijų turinį, rengti ir bendrinti dokumentus bendradarbiaujant su kitais

Taksonomijos lygmenys	Besimokančiųjų atliekami veiksmai (Anderson ir Krathwohl, 2001)	Mokymosi veiklos, naudojant IKT (Churches, 2009)
Analizė	Palyginti, organizuoti, išskaidyti, suskirstyti, apibrėžti, išvesti, struktūruoti, integruoti	Integruoti į vieną keletą skirtingų duomenų šaltinių (anglų k. <i>mashing – mash ups</i>), kurti informacijos saitus (anglų k. <i>linking</i>)
Vertinimas	Tikrinti, daryti prielaidas, kritikuoti, eksperimentuoti, nuspręsti, testuoti, nustatyti, stebėti, diskutuoti	Bendradarbiauti ir kurti tinklą (anglų k. <i>networking</i>), testuoti programas, procesus ir procedūras, dalyvauti pokalbiuose ir diskusijose, vesti diskusijas ir forumus, vertinti komentarus, komentuoti tinklaraščius, užtikrinti informacijos patikimumą, validumą (anglų k. <i>Validating Information Rubric</i>), tvarkyti bibliografiją
Kūryba	Projektuoti, konstruoti, planuoti, gaminti, sukurti, išrasti	Programuoti, kurti interaktyviųjų medijų turinį, projektuoti procesus, kurti teksto ir vaizdo tinklaraščius, kurti vikį, interaktyviuosius žaidimus, interaktyvias reklamas, skelbti, platinti ir susikurti naujienas bei pranešimus (anglų k. <i>podcasting, videocasting, broadcasting</i>), suburti nuolatinių prenumeratorių ratą

Bloomo taksonomija įvairiomis formomis atspindi mokymosi procesą ir iš esmės parodo, kaip mes mokomės. „Kad galėtume suprasti sąvoką, idėją ar reiškinį, turime žinoti apie jį ir jį prisiminti. Prieš vartodami sąvokas ar taikydami idėjas, turime jas suprasti. Norėdami analizuoti, turime suprasti ir pritaikyti. Norėdami įvertinti, turime gebėti analizuoti, o, norėdami ką nors sukurti, turime ir suprasti, ir gebėti pritaikyti, ir analizuoti, ir įvertinti“ (Churches, 2009). Laikantis tradicinės koncepcijos, ugdymo procesas pradamas nuo paprastesnių tikslų ir einama prie sudėtingesnių. Šiuo metu diskutuojama ir apie „apverstą“ sistemą, kai visų kitų tikslų siekiama per kūrybą ir projektinę veiklą, tačiau galimybės įgyvendinti mokymąsi analizuoti, kurti ir vertinti, t. y. aukštesnius pažinimo tikslus, įvairuoja priklausomai nuo studijuojamo dalyko specifikos. Yra dalykų, kuriems nebūtina pasiekti aukščiausiąjį, kūrybos, lygmenį (pvz., programavimo kalbos sintaksės žinios), tačiau, remiantis A. Churchesu (2009), programavimo veikla plačiąja prasme vis dėlto priskiriama kūrybos lygmeniui (Churches, 2009). Vadinasi, ugdant programavimo kompetenciją, teks „pereiti“ visus Bloomo taksonomijos lygmenis. Penkių mokslininkų grupė išanalizavo ir 2008 metais paskelbė, kaip taikyti Bloomo taksonomiją programavimo gebėjimams vertinti (Thompson, Luxton-Reilly, Whalley, Hu, Robbins, 2008). Kiekvienam lygmeniui buvo suformuluoti dalyko rezultatai ir numatyti mokymosi metodai (žr. 6 lentelę).

**Atnaujintos Bloomo taksonomijos taikymas mokantis programavimo
(Thompson ir kt., 2008)**

Atnaujintos Bloomo taksonomijos lygmenys	Pasiektas dalyko rezultatas	Taikomi mokymosi metodai
Įsiminimas	Žino pagrindines programavimo sąvokas, programavimo kalbos leksiką ir sintaksės reikalavimus, programos tekste atpažįsta tam tikrus programinius konstruktus	Paskaita, vizualizavimas, vaizdo įrašai, garso įrašai, pavyzdžiai, iliustracijos analogijos, informacijos paieška ir kt.
Supratimas	Supranta ir gali paaiškinti pagrindines programavimo sąvokas, geba žodžiu nusakytą algoritmą užrašyti programavimo kalba arba pavaizduoti grafiškai, pateikia sąvokas iliustruojančius pavyzdžius, geba ištaisyti sintaksės klaidas, paaiškinti tipinių algoritmų veikimą	Klausimai, diskusijos, pristatymai, rašto darbai, refleksija, minčių lietus, sąvokų žemėlapis ir kt.
Taikymas	Geba panaudoti įgytas žinias, atpažįsta panašias situacijas ir taikydamas tipinius algoritmus bei remdamasis turimais pavyzdžiais rašo nedidelės apimties programėles	Praktinės užduotys, demonstracijos, simuliacijos, vaidmenų žaidimai, projektavimas, planavimas ir kt.
Analizė	Geba skaidyti užduotį į sudėtinę dalis, analizuoti parašytas programas, demonstruoja programų derinimo įgūdžius (pvz.: identifikuoja algoritmo logines klaidas ir jas ištaiso, pasirenka tinkamus duomenų tipus ir savo pasirinkimą pagrindžia, identifikuoja nesvarbius komponentus, numato programos tobulinimo kryptis ir pan.)	Probleminės užduotys, atvejo analizė, diskusijos, klausimai ir kt.
Vertinimas	Geba įvertinti ar parašyta programa tenkina numatytus reikalavimus, parenka tinkamas testuoti duomenų reikšmes, kritiškai vertina parašytas programas, remdamasis programavimo standartais arba sprendžiamos probleminės srities kokybiniais kriterijais	Kritinis mąstymas, atvejo analizė, vertinimas ir įsivertinimas ir kt.
Kūryba	Geba kurti naujus alternatyvius algoritmus arba jų kombinacijas pažįstamoms ir naujoms sudėtingesnėms problemoms spręsti	Projektai, probleminės užduotys, kūrybinės užduotys, planavimas (vystymas) ir kt.

Ugdant pradedančiųjų programavimo gebėjimus, laikantis tradicinės koncepcijos, pagrindinis dėmesys turi būti skiriamas pirmosioms pažintinių tikslų pakopoms: įsiminimo, supratimo, taikymo ir analizės. Ar būtina nuosekliai laikytis Bloomo taksonomijos? Tikrai ne, mokymasis gali prasidėti nuo bet kurio lygmens – ugdytojas turi atsižvelgti į skirtingą besimokančiųjų patirtį.

2. STUDENTŲ PROGRAMAVIMO GEBĖJIMŲ UGDYMO SI, ĮVERTINANT MOKYMO SI STILIŲ IR MOKYMO SI METODŲ DERMĖ, TYRIMO ORGANIZAVIMAS IR METODOLOGIJOS PAGRINDIMAS

Studentų programavimo gebėjimų ugdymosi, įvertinant mokymosi stilių ir mokymosi metodų dermę, tyrimo metodologijos pagrindas – pažinimo proceso konstruktyvistinė prigimtis, mokymosi kaip individualių pažinimo gebėjimų konstravimas ir besimokančiojo patirčių svarba ugdymosi procese. Pasirinkta kokybinių tyrimų strategija sudarė prielaidas tyrėjai gauti išsamią ir reikšmingą informaciją apie tyrimo dalyvių patirtis ugdantis programavimo gebėjimus. Teorinės pažinimo reiškinių aiškinančios nuostatos yra šios:

1. Pragmatizmo filosofinė teorija – individo mąstymas siejamas su veiklos būdais, o pagrindinis elementas laikomas patyrimas kaip individualios tiesos pažinimo priemonė (Džeimsas, 1995; Dewey, 2004, 2013).
2. Konstruktyvizmo teorija ir jos taikymas ugdymo procese – akcentuojama, kad mokymasis yra aktyvus ir konstruktyvus procesas (Brooks, 1999; Ramsden, 2000; Windschitl, 2002; Gonzales, 2004; Grandy, 2010 ir kt.). Kiekvieno besimokančiojo žinių ir patirties rezultatas – individuali konstrukcija, todėl besimokantieji nepasyviai perima informaciją, o aktyviai įgyja žinių ir formuojasi įgūdžius, remdamiesi jau turimomis žiniomis, siedami jas ir sąveikaudami su aplinka (Martišauskienė, 2008; Bitinas, 2013).
3. Ugdymosi paradigmų kaita – nuo mokymo pereinama prie mokymosi. Mokymasis suprantamas kaip aktyvus konstravimo procesas, orientuotas į tikslą, paremtas individualiu informacijos interpretavimu ir bendradarbiavimu, kai yra siejamas su kontekstu ir aplinka (Argyris, Schön, 1974; Bitinas, 2000).
4. Laisvojo ugdymo paradigma – pedagoginės technologijos orientuojamos į ugdytinių vidinių galių, saviraiškos, saviraidos, savisklaidos galimybių plėtrą ir siekiama, kad ugdymo turinys tenkintų besimokančiojo interesus (Bitinas, 2000; Bruzgelevičienė, 2008).
5. Patirtinio mokymosi teorija – siejama su ugdomosios veiklos tobulinimo tyrimu (Kolb, 1984; 200; 2005) ir pasirinkta P. Honey ir A. Mumfordo mokymosi stilių tipologija (Honey, Mumford, 2000), pagal kurią kiekvienas mokymosi stiliaus tipas asocijuojamas su pirmenybe konkrečiam patirtinio mokymosi ciklo etapui. Kiekvienas mokymosi stilius turi privalumų ir trūkumų; jie gali

būti svarbūs vienoje situacijoje ir visiškai nesvarbūs kitoje (Honey, Mumford, 2000).

Studentų programavimo gebėjimų ugdymosi, įvertinant mokymosi stilių ir mokymosi metodų dermę, tyrimo metodologija pateikiama šia tvarka: aprašoma tyrimo konstravimo logika, vaizduojama tyrimo struktūra, aprašoma MSMMD modelio kūrimo eiga, žvalgomojo tyrimo ir veiklos tyrimo metodikos.

2.1. TYRIMO ORGANIZAVIMAS IR ATLIKIMAS

Rengiantis tyrimui ir formuluojant tyrimo problemą atlikta sisteminė mokslinės literatūros analizė, derinti informacijos paieškos, sisteminimo, lyginamosios analizės ir apibendrinimo metodai. Remiantis analizės rezultatais suformuluotas teorinis ir metodologinis disertacinio darbo pagrindimas. Apibendrinus teorines mokymosi stilių taikymo ugdymosi procese ir programavimo gebėjimų ugdymosi prielaidas, sudarytas MSMMD modelis, kurio paskirtis – ugdyti studentų programavimo gebėjimus. Siekiant įvertinti šio modelio taikymo ugdymosi praktikoje galimybes suplanuota atlikti žvalgomąjį tyrimą, o norint verifikuoti modelį – veiklos tyrimą. Disertacinio tyrimo dizaino schema pateikta 6 paveiksle.

Teorinis ir empirinis tyrimas atliktas keliais etapais.

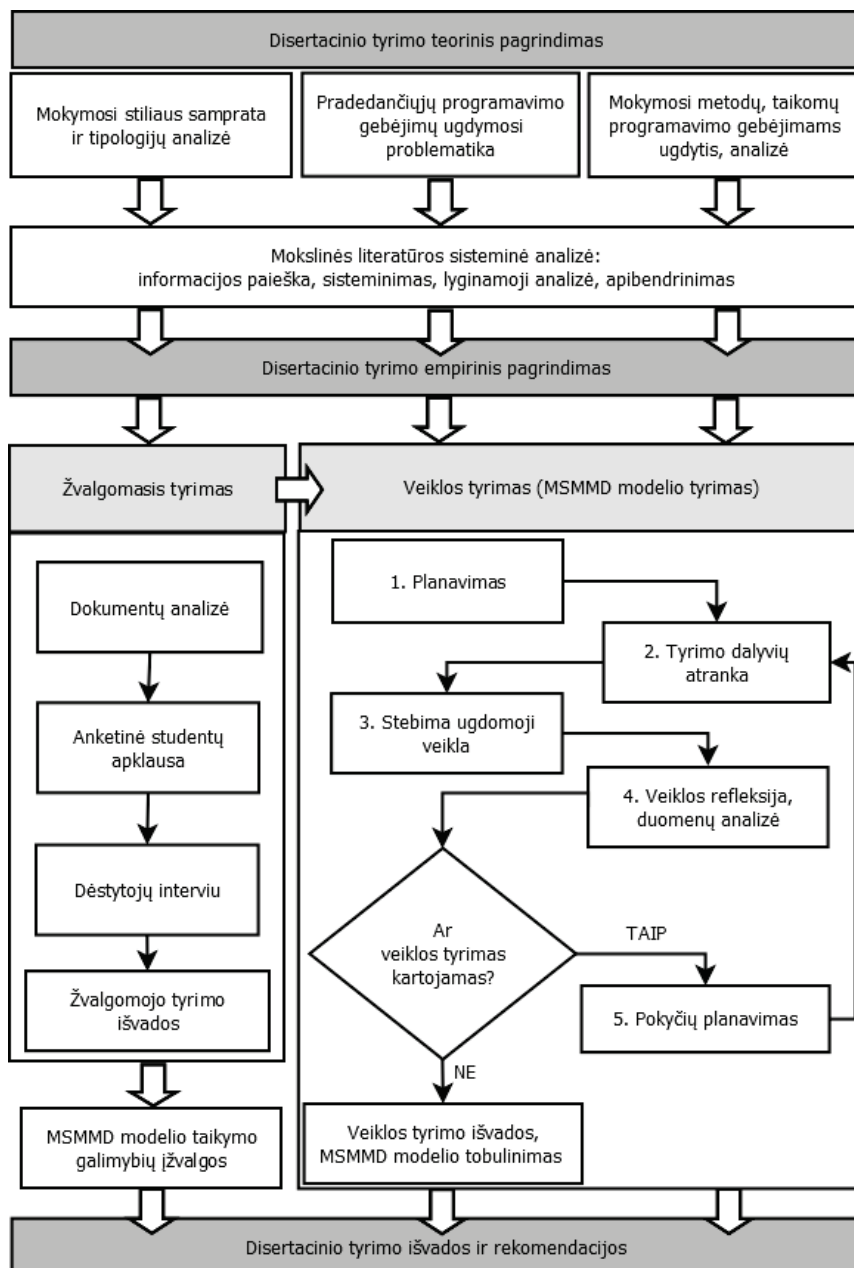
I tyrimo etapas – mokslinės literatūros ir dokumentų analizė. Analizuota Lietuvos ir užsienio autorių mokslinė literatūra, apimanti mokymosi stilių tipologijų, programavimo gebėjimų ugdymosi ir šiems gebėjimams ugdytis taikomų mokymosi metodų klausimus. I etapo rezultatai:

1. Atskleistos konceptų *mokymosi stilius* ir *pradedančiųjų programavimo gebėjimai* sampratos.
2. Išryškinta pradedančiųjų programavimo gebėjimų ugdymosi problematika ir šiems gebėjimams ugdytis taikomų mokymosi metodų specifika. Mokymosi metodų sąrašas pateiktas 9 priede.

II tyrimo etapas – modelio sudarymas, tyrimo priemonių ir ĮPK programos parengimas. Šiame etape buvo atlikti pasirengimo veiklos tyrimui darbai. II tyrimo etapo rezultatai:

1. Parengtas MSMMD modelis (žr. 10 priedą), skirtas pradedančiųjų programavimo gebėjimams ugdytis.
2. Išversta ir aprobuota P. Honey ir A. Mumfordo mokymosi stilių klausimyno lietuviškoji versija (žr. 11 priedą).

3. Parengta ir įvertinta įvadinio programavimo kurso (ĮPK) programa, skirta koleginiams informatikos krypties studijoms (žr. 6 priedą).



6 pav. Mokslinio tyrimo struktūrinė schema

III tyrimo etapas – žvalgomasis tyrimas. Šiame etape buvo naudojamos kiekybinių ir kokybinių tyrimų strategijos. Dokumentų lyginamoji analizė apėmė 37 ĮPK programas ir vieną CSC2013 programą. Anketinės studentų apklausos imtis – 377 informatikos, informacijos sistemų, programų sistemų ir informatikos inžinerijos krypties studijų studentai. Interviu su dėstytojais imtis – 5 informantai. III etapo rezultatai:

1. Atskleista, kaip Lietuvos aukštųjų mokyklų įvadinio programavimo kursuose organizuojamas pradedančiųjų programavimo gebėjimų ugdymo procesas.
2. Apibendrintos sudaryto MSMD modelio taikymo Lietuvos aukštųjų mokyklų įvadinio programavimo kursuose išvagos.

IV etapas – veiklos tyrimas. Tai kokybinis tyrimas, kartotas tris kartus. Tyrimo imtis – po 7 studentus kiekvienam kartotiniam ciklui. IV etapo rezultatai:

1. Išryškinti programavimo gebėjimų ugdymo proceso ypatumai. Atskleista mokymosi stilių ir mokymosi metodų dermės raiška pradedančiųjų programavimo gebėjimų ugdymo procese.
2. Remiantis veiklos tyrimo rezultatais patobulintas MSMD modelis.

2.2. MOKYMO SI STILIŲ IR MOKYMO SI METODŲ DERMĖS MODELIS PRADEDANČIŲJŲ PROGRAMAVIMO GEBĖJIMAMS UGDYTI

Mokymosi stilių nustatymo priemonių validumas ir patikimumas. Kuriant mokymosi stilių ir mokymosi metodų dermės modelį ir renkant jo tipologiją, buvo atsižvelgiama į mokymosi stiliaus nustatymo priemonių (klausimynų) patikimumą (anglų k. *reliability*). Patikimumas vertinamas atsižvelgiant į šias pagrindines charakteristikas (Norušis, 2005; Garson, 2009, cituojama Pukėnas, 2009):

- klausimyno skalės vidinį suderinamumą (anglų k. *scale internal consistency*), kuriam įvertinti įprastai yra skaičiuojamas *Cronbach alpha*³ (α) koeficientas;
- klausimyno patikimumą kartotinių tyrimų atžvilgiu (anglų k. *test-retest reliability*), kuris remiasi dviejų (ar daugiau) bandymų koreliacija;
- vertinimo patikimumą (anglų k. *inter-rater reliability*), kuris remiasi koreliacija tarp dviejų (ar daugiau) ekspertų vertinimų.

³ Socialiniuose tyrimuose *Cronbach alpha* koeficientas laikomas patikimu, kai jo reikšmė – apie 0,7, ir nepriimtina, kai jo reikšmė – mažesnė už 0,5 (Gliem, Gliem, 2003).

D. Kolbo klausimynas *The Learning Style Inventory* (LSI) buvo sukurtas 1969 metais, paskui keletą kartų tobulintas (LSI-1, 1971, 1976; LSI-2, 1985; LSI-2, 1993; Kolb LSI version 3, 1999; Kolb LSI version 3.1, 2005; Kolb LSI version 4.0, 2011). Kaip jau minėta pirmajame skyriuje, D. Kolbas patirtinio mokymosi teorijoje išskiria individualius mokymosi ir informacijos suvokimo būdus: vieni besimokantieji informaciją suvokia per pojūčius, remdamiesi konkrečiomis patirtimis, kiti – sistemindami, analizuodami (Kolb, 1984). Pirmąją klausimyno versiją sudarė 9 diagnostiniai kintamieji, glaudžiai susiję su patirtinio mokymosi teorija. Kiekviename iš 9 klausimų respondentams pateikiama po 4 teiginius, kuriuos reikia suranguoti: nuo 4 – teiginys, kuris labiausiai priimtinas; iki 1 – teiginys, su kuriuo mažiausiai sutinkama. Kiekvienoje teiginių grupėje turi būti nurodyti visi prioritetai, tai yra nuo 4 iki 1. LSI psichometrinių tyrimų (Atkinson, 1991; Cornwell, Manfredo, 1991, 1994; Geiger, Boyle, Pinto, 1993; Stout, Ruble, 1991; Veres, 1991, cituojama Kolb, 2005) rezultatai rodė prastą klausimyno patikimumą. Patobulintą versiją (LSI-2) sudaro dvylika teiginių, kurių kiekvienas turi keturias alternatyvias pabaigas; jas respondentai turi suranguoti nuo 1 iki 4 (Kolb, 2005). Naujausių tyrimų duomenimis, LSI-2 klausimynui nustatytas geras vidinis suderinamumas. *Cronbach alpha* koeficiento reikšmės visose pozicijose didesnės arba siekia 0,77. Kūrėjų teigimu, naujos klausimyno versijos visi psichometriniai rodikliai yra geresni (Brew, 2002; Wierstra, de Jong, 2002; Kayes, 2005 cituojama Kolb, 2005).

P. Honey ir A. Mumfordo klausimynas *The Learning Styles Questionnaire* (LSQ) pirmą kartą publikuotas 1982 metais. Šiuo metu egzistuoja dvi klausimyno versijos. Klausimyno teiginiai apibūdina besimokančiojo elgseną. Originaliąją klausimyno versiją sudaro 4 grupės po 20 teiginių, apibūdinančių kiekvieną mokymosi stilių (aktyvistas, mąstytojas, teoretikas, pragmatikas). Teiginiai klausimyne pateikiami nesugrupuoti, naudojama dichotominė skalė: sutinku, nesutinku. Nustatant vyraujančią mokymosi stilių skaičiuojama kiekvienos grupės patvirtintų teiginių suma. 2000 metais parengta sutrumpinta klausimyno versija, kurią sudaro 40 teiginių, po 10 kiekvienam mokymosi stilių tipui. Mokslinių tyrimų (B. J. Tepper ir kt. (1993) ir A. Duff, T. Duffy (2002)) duomenimis, gauti panašūs rezultatai: *Cronbach alpha* koeficientų reikšmės svyruoja nuo 0,52 iki 0,76. S. M. DeCiantis ir M. J. Kirtonas (1996) nustatė pakankamą vidinį suderinamumą visoms keturioms klausimyno dalims (aktyvistas – 0,76; mąstytojas – 0,76; teoretikas – 0,67; pragmatikas – 0,64). N. Van Zwanenbergas ir kt. (2000) gavo mažesnes *Cronbach alpha* koeficiento reikšmes (aktyvistas – 0,74; mąstytojas – 0,68; teoretikas – 0,64; pragmatikas – 0,59). G. E. Pickworthas ir kt. (2000), pristatydami LSQ patikimumo analizę, teigė, kad gautos *Cronbach alpha* koeficientų reikšmės rodo gana aukštą visų keturių klausimyno skalių vidinį suderinamumą (Pickworth ir kt., 2000).

Pirmoji R. M. Felderio ir L. K. Silverman klausimyno *Index of Learning Styles* (ILS) versija sukurta 1991 metais. Klausimyno kūrėjai – R. M. Felderis ir B. A. Solomon. Elektroninė klausimyno versija sukurta ir paskelbta internete po šešerių metų, 1997-aisiais. ILS klausimyną sudaro 44 diagnostiniai kintamieji, po 11 kiekvienai iš keturių teorijoje apibrėžtų dimensijų: sensoriniai-intuityvūs; vizualiniai-verbaliniai; aktyvūs-svarstantieji; nuoseklieji-globalieji. 1994 metais atlikta klausimyno faktorinė analizė, jos metu išryškinti statistiškai nereikšmingi diagnostiniai kintamieji atmesti ir pakeisti naujais. Klausimyno patikimumą tyrė kelios mokslininkų grupės: N. Seery ir kt. (2003 cituojama Felder, Spurlin, 2005) apklausė tuos pačius tiriamuosius po 4 savaitių; G. A. Livesay ir kt. (2002 cituojama Felder, Spurlin, 2005) ir M. S. Zywno (2003 cituojama Felder, Spurlin, 2005) kartojo tyrimą po 7–8 mėnesių. Galop nustatyta, kad testas išoriškai patikimas, mat visų keturių klausimyno skalių koreliacijos koeficientai kinta nuo 0,7 iki 0,9, kai testavimas kartotas po 4 mėnesių, o kai po 7–8 mėnesių, – nuo 0,5 iki 0,8. Apskaičiuotos vidinio suderinamumo *Cronbach alpha* koeficientų reikšmės trijuose tyrimuose kiek didesnės už 0,5.

Kaip jau minėta anksčiau (1.1 skyriuje), D. Kolbo bei P. Honey ir A. Mumfordo mokymosi stilių tipologijos yra panašios, nes abi remiasi ta pačia D. Kolbo patirtinio mokymosi teorija. Pasak C. W. Allinsono ir J. Hayes (1988), LSQ klausimynas geresnis, kadangi juo vertinamas besimokančiojo elgesys, o ne mokymosi pasirinkimai ar motyvai (Allinson, Hayes, 1988). LSI ir LSQ klausimynų patikimumas ir validumas lygintas ir vertintas daugelio tyrėjų (Allinson, Hayes, 1988; Sims, Veres, Shake, 1989; Swailes, Senior, 1999; Sadler-Smith, 2001; Duff, Duffy, 2002; Kappe, Boekholt, Den Rooyen, Van der Flier, 2009; Fleming, Mckee, Huntley-Moore, 2011; Manolis, Burns, Assudani, Chinta, 2013). Lyginant LSI ir LSQ kartotinius tyrimus stabilesni rezultatai gaunami vertinant LSQ klausimyną (Kappe, Boekholt, Den Rooyen, Van der Flier, 2009; Manolis, Burns, Assudani, Chinta, 2013). N. Van Zwanenbergas, L. J. Wilkinsonas, A. Andersonas, (2000), palyginę ILS ir LSQ, pateikė išvadą, kad LSQ yra geresnio vidinio suderinamumo klausimynas. Tačiau, autorių teigimu, kai kurie abiejų priemonių psichometriniai rodikliai yra ginčytini (Van Zwanenberg, Wilkinson, Anderson, 2000).

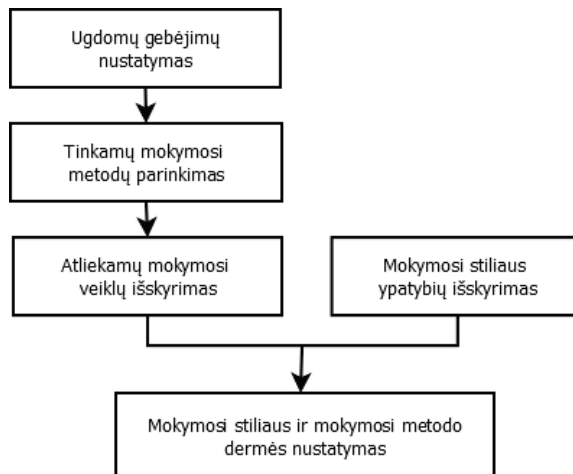
Apibendrinant galima teigti, kad mokymosi stilių nustatymo klausimyno (LSQ) patikimumo ir validumo tyrimų rezultatai yra geresni lyginant juos su kitų analizuotų klausimynų patikimumo tyrimais.

Modeliavimas. Mokslinis modeliavimas – tai vieno objekto savybių pakartojimas kitame objekte (modelyje), norint geriau pažinti pirmąjį objektą. Egzistuoja daug bei įvairių metodų ir teorijų apie įvairių rūšių mokslinį modeliavimą. Mokslinio modeliavimo teorija yra pagrįsta filosofijos, sistemų ir žinių vizualizavimo teorijomis.

Įvertinus dėstyto patirtį ir mokslinės literatūros analizės apie mokymosi stilių naudojimą ugdymosi praktikoje, nuspręsta sukurti MSMMD modelį, skirtą pradedančiųjų programavimo gebėjimams ugdytis. Modeliui sudaryti buvo pasirinkta reikiama mokymosi stilių tipologija remiantis atlikta mokymosi stilių teorijų analize. Pasirinkimą lėmė teorijos kūrėjų apibrėžiama mokymosi stiliaus samprata – ji susijusi su pageidaujamų veiklos būdų mokantis pasirinkimu, o ne su besimokančiųjų asmens charakteristikomis. Tokiam modeliavimui tinkamos D. Kolbo (1984), P. Honey ir A. Mumfordo (1982) bei R. M. Felderio ir L. K. Silverman (1988) mokymosi stilių tipologijos. Informatikos inžinerijoje tai dažniausiai adaptyvioms ir rekomendacinėms mokymosi aplinkoms modeliuoti pasirenkamos mokymosi stilių tipologijos (Hawk, Shah, 2007; Urbonienė, 2013).

MSMMD modeliui kurti pasirinkta P. Honey ir A. Mumfordo mokymosi stilių tipologija ir dažniausiai pradedančiųjų programavimo gebėjimams ugdytis pasitelkiami mokymosi metodai (žr. 9 priedą). Modeliavimo eigą galima pavaizduoti grafiškai išskiriant tokius procesus (žr. 7 paveikslą):

- ugdomų gebėjimų nustatymas;
- tinkamų mokymosi metodų parinkimas;
- besimokančiojo atliekamų mokymosi veiklų taikant metodą identifikavimas;
- būdingų mokymosi stiliaus ypatybių išskyrimas;
- mokymosi veiklų ir būdingų ypatybių derinimas.



7 pav. MSMMD modeliavimo procesas

Pasak P. Honey ir A. Mumfordo (1982), mokymosi stilius – tai asmens nuostatos ir elgsena, kurie nulemia pageidaujamų mokymosi būdų pasirinkimą. Autoriai pabrėžia, kad „nė vienas stilius nėra geresnis už kitą. Kiekvienas iš jų turi stipriąją ir silpnąją puses, o jų stiprybės gali būti svarbios vienoje situacijoje ir visiškai nesvarbios kitoje. Be to, mokymosi stilius yra tik vienas iš daugelio veiksnių, darančių įtaką asmens mokymosi procesui“ (Honey, Mumford, 2000). Jie išskyrė keturis mokymosi stilių tipus ir pateikė pagrindines jų ypatybes (žr. 7 lentelę):

- **aktyvistai** (A) pirmenybę teikia aktyviai veiklai;
- **mąstytojai** (M) stebi ir apmąsto tai, kas įvyko;
- **teoretikai** (T) stengiasi suprasti priežastis, sąvokas, sąryšius;
- **pragmatikai** (P) mėgsta praktiškai išbandyti teorijas, idėjas ir metodus.

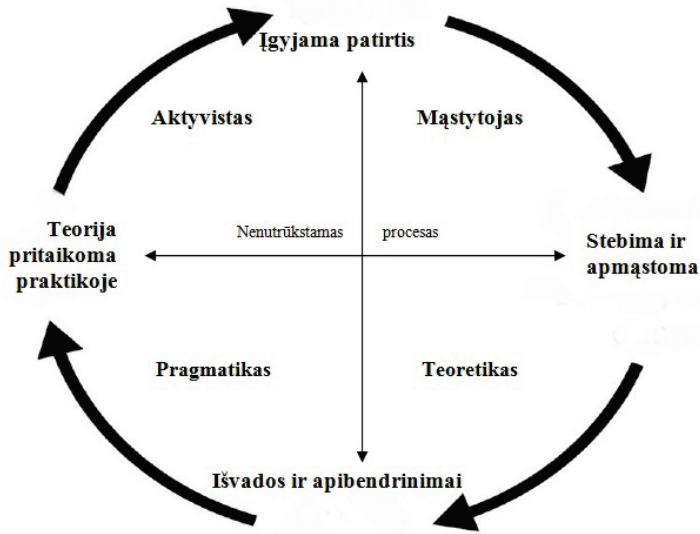
7 lentelė

Mokymosi stiliams būdingos ypatybės (Honey, Mumford, 2000)

Būdingos ypatybės			
Aktyvistai A	Mąstytojai M	Teoretikai T	Pragmatikai P
Skiriamasis bruožas – būtinybė veikti, daryti, įgyvendinti; organizuoti veiklas, susidoroti su probleminėmis situacijomis, reikalaujančiomis greitos adaptacijos prie besikeičiančių aplinkybių; daugiau remiasi intuicija negu logika, problemas linkę spręsti intuityviai mėgindami ir klysdami; naudojasi kitų sukurtomis teorijomis ir analizių rezultatais, todėl mėgsta dirbti komandoje; labiausiai linkę rizikuoti; moka bendrauti su žmonėmis, tačiau kartais gali pasirodyti nekantrūs, reiklūs	Pirmenybę teikia konkrečiai patirčiai ir reflektavimui; geba išsivaizduoti, įvertinti konkrečias situacijas iš įvairių perspektyvų; plati vaizduotė, geba numatyti vizijas, improvizuoti situacijose, reikalaujančiose rasti idėjų; noriai bendrauja su žmonėmis, domisi žmonių tarpusavio santykiais, yra emocionalūs	Pirmenybę teikia reflektavimui ir abstrakčiam konceptualizavimui; lengvai kuria teorinius modelius, sieja turimą informaciją ir abstrakčius situacijos elementus į visumą; linkę daryti induktyvias išvadas, apibendrinti rezultatus; mieliau dirba su abstrakčiomis idėjomis ir koncepcijomis nei su žmonėmis	Transformuoja teoriją, norėdami pasirengti konkrečiai veiklai; plėtoja idėjas; linkę daryti hipotetines-dedukcines išvadas; domisi daiktais ir teorijomis (kurias mėgsta patikrinti), specifinių problemų sprendimu; mažai domisi žmonėmis, nėra emocionalūs

Autorių apibrėžtų mokymosi stilių pagrindas – pirmenybė konkrečiam mokymosi ciklo etapui (žr. 8 paveikslą). Nuo mokymosi stiliaus priklauso besimokančiajam būdingos ypatybės (žr. 7 lentelę), jo atliekama mokymosi veikla, taikomi mokymosi

metodai, keliami klausimai, bendravimas, apibendrinimas. Planuojant mokymą kiekvienam mokymosi stiliaus tipui, atsižvelgiant į konkrečias jo ypatybes, rekomenduojamos tinkamos mokymosi veiklos (anglų k. *learning activities*). P. Honey ir A. Mumfordo pasiūlyta mokymosi veiklų klasifikacija atitinkamiems mokymosi stiliams pateikiama 8 lentelėje.



8 pav. Mokymosi stilių tipologija (Honey, Mumford, 2000)

8 lentelė

Mokymosi stilių ir mokymosi veiklų sąsajos (Honey, Mumford, 2000)

Pageidaujamos mokymosi veiklos			
Aktyvistai A	Mąstytojai M	Teoretikai T	Pragmatikai P
Aktyvus mokymasis; verslo žaidimų simuliacija; darbų kaita; diskusijos mažose grupėse; vaidmenų žaidimai; kitų mokymas; mokymasis lauke (ne klasės aplinkoje)	E. mokymasis; mokymosi peržiūra (apmąstymas); paskaitų klausymas, prezentacijos; vaidmenų žaidimo stebėjimas; skaitymas; savarankiškas mokymasis	Analitinė veikla; pratimai su teisingais atsakymais; paskaitų klausymas; savarankiškas mokymasis; individualūs pratimai; vaizdo pamokos	Aktyvus mokymasis; probleminės diskusijos; diskusijos mažose grupėse; problemų sprendimas; grupinio darbo taikomosios užduotys; projektinis darbas

Remiantis P. Honey ir A. Mumfordo aprašytais mokymosi stiliaus atstovų ypatybėmis ir mokymosi veiklų klasifikacija pagal mokymosi stilius, sudaryta mokymosi metodų, skirtų programavimo gebėjimams ugdytis, klasifikacija. Metodų klasifikacija pagal mokymosi stilius pateikta 9 lentelėje.

9 lentelė

Mokymosi stilių ir mokymosi metodų klasifikacija

Mokymosi stilius	Būdingos savybės	Mokymosi metodai ir būdai
Aktyvistas	Būtinybė veikti, daryti, įgyvendinti; naujų patirčių paieška; veiklų organizavimas	MM12: Minčių lietus; MM19: Pristatymai; MM22: Programavimas porose; MM20: Žaidimų programavimas; MM21: Varžybos
	Problemas linkęs spręsti intuityviai, mėginamas ir klysdamas	MM05: Problemų sprendimas taikant „mėginimų ir klaidų“ metodą
	Mėgsta dirbti komandose	MM04: Darbas grupėse bendradarbiaujant; MM07: Diskusija; MM08: Programavimas porose; MM09: Durstinys
Mąstytojas	Pirmenybę teikia konkrečiai patirčiai ir reflektiviajam stebėjimui	MM02: Aiškinimas; MM06: Demonstravimas; MM13: Paskaita; MM14: Lektūra; MM18: Vaizdo pamokos
	Geba įsivaizduoti, įvertinti konkrečias situacijas iš įvairių perspektyvų, improvizacija situacijose, reikalaujančiose rasti naujų idėjų	MM01: Algoritmų ir programų vizualizavimas; MM12: Minčių lietus; MM17: Sąvokų žemėlapis
	Noriai bendrauja su žmonėmis, domisi žmonių tarpusavio santykiais, yra emocionalūs	MM04: Darbas grupėse bendradarbiaujant; MM07: Diskusija; MM08: Programavimas porose; MM09: Durstinys
Teoretikas	Pirmenybę teikia reflektiviajam stebėjimui ir abstrakčiajam konceptualizavimui	MM02: Aiškinimas; MM06: Demonstravimas; MM13: Paskaita; MM14: Lektūra; MM18: Vaizdo pamoka
	Mieliau dirba su abstrakčiomis idėjomis ir koncepcijomis nei su žmonėmis	MM19: Vikis; MM11: Individualus darbas
	Lengvai kuria teorinius modelius, sieja turimą informaciją ir abstrakčius situacijos elementus į visumą, linkęs daryti induktyvias išvadas, apibendrinti rezultatus	MM01: Algoritmų ir programų vizualizavimas; MM17: Sąvokų žemėlapis; MM21: Problemų sprendimas

Mokymosi stilius	Būdingos savybės	Mokymosi metodai ir būdai
Pragmatikas	Transformuoja teoriją, norėdamas pasirengti konkrečiai veiklai; plėtoja idėjas	MM21: Problemų sprendimas MM07: Diskusija (probleminė); MM20: Žaidimų programavimas
	Domisi daiktais ir teorijomis (kurias mėgsta patikrinti), kaip išspręsti specifines problemas	MM03: Darbas pagal instrukcijas „žingsnis po žingsnio“; MM10: Programos teksto skaitymas ir veikimo sekimas; MM06: Demonstravimas; MM16: Savikontrolės testai
	Mažai domisi žmonėmis, nėra emocionalūs	MM11: Individualus darbas; MM19: Vikis

P. Honey ir A. Mumfordo mokymosi stilių tipologijoje atsižvelgiama į besimokančiojo poreikius, o mokymosi procesas vertinamas iš besimokančiojo pozicijų. Tai labai svarbu siekiant paisyti individualių besimokančiųjų savybių. Rengiant MSMD modelį, skirtą programavimo gebėjimams ugdytis, panaudota P. Honey ir A. Mumfordo mokymosi stilių tipologija ir disertantės dėstymo praktikoje taikytas J. T. Mayes, C. J. Fowlerio mokymosi ciklas su grįžtamuoju ryšiu. Mokymosi stilių ir mokymosi metodų dermės modelis pateiktas 10 lentelėje.

10 lentelė

Mokymosi stilių ir mokymosi metodų dermės modelis

Konceptualizacija				
Mokymosi metodai	A	M	T	P
MM01: Algoritmų ir programų vizualizavimas		X	X	
MM02: Aiškinimas		X	X	
MM06: Demonstravimas		X	X	X
MM07: Diskusija	X	X		X
MM09: Durstinys	X	X		
MM10: Programos teksto skaitymas ir veikimo sekimas				X
MM12: Minčių lietus	X	X		
MM13: Paskaita			X	
MM14: Lektūra		X	X	
MM15: Pristatymai	X			
MM16: Savikontrolės testai				X
MM17: Sąvokų žemėlapis		X	X	
MM18: Vaizdo pamoka		X	X	
MM19: Vikis			X	X

Konstravimas				
Mokymosi metodai	A	M	T	P
MM03: Darbas pagal instrukcijas „žingsnis po žingsnio“				X
MM04: Darbas grupėse bendradarbiaujant	X	X		
MM05: Problemų sprendimas	X		X	X
MM08: Programavimas porose	X	X		
MM09: Durstinys	X	X		
MM11: Individualus darbas			X	X
MM20: Žaidimų programavimas	X			X
MM21: Varžybos	X			
Dialogas				
Mokymosi metodai	A	M	T	P
MM07: Diskusija	X	X		X
MM16: Savikontrolės testas				X
MM11: Individualus darbas			X	X

Modelio esmė: parinkti skirtingų mokymosi stilių atstovams (aktyvistui, mąstytojui, teoretikui, pragmatikui) tinkamus mokymosi metodus kiekviename mokymosi ciklo (konceptualizacijos, konstravimo, dialogo) etape. P. Honey ir A. Mumfordo mokymosi stilių tipologijoje pateikiama mokymosi veiklų klasifikacija pagal mokymosi stilių tipus, tačiau įgytų žinių ir gebėjimų vertinimo metodų šioje tipologijoje nedaug. Kadangi J. T. Mayeso, C. J. Fowlero apibrėžtas dialogo etapas labiau sietinas būtent su įgytų žinių ir gebėjimų vertinimu, tai ir mokymosi veiklų šiam etapui galima parinkti tik keletą. Įgytoms žinioms ir gebėjimams vertinti pasitelkta Bloomo taksonomija.

Kalbant apie konkretų besimokantįjį, pateiktas metodų skirstymas pagal mokymosi stilius nėra absoliutus. Besimokančiajam paprastai yra būdingi keli mokymosi stiliai, tačiau vienas iš jų yra labiau išreikštas. Metodas gali būti patrauklus ir dėl dėstytojo išradinumo. Vis dėlto studijų procese naudojamas MSMMD modelis sudarys daugiau galimybių individualizuoti ar net personalizuoti mokymą. Norint naudoti modelį nuolatinių studijų procese rekomenduojama pasitelkti virtualiąją mokymosi aplinką ir numatyti daugiau laiko individualioms konsultacijoms. Besimokančiajam priimtinausius mokymosi metodus gali parinkti dėstytojas, automatizuota adaptyvi mokymosi aplinka arba pats besimokantysis. Galimi MSMMD naudojimo atvejai:

1. Rengiant mokymosi stilių atitinkančius mokymosi scenarijus (pvz., planuojant darbą grupėse įvairių stilių atstovams numatyti skirtingas ir jų mokymosi stilių atitinkančias mokymosi veiklas ir pan.).

2. Kuriant adaptyvias ar rekomendacines programavimo mokymosi aplinkas, kuriose besimokančiajam mokymosi veiklas automatizuotai parenka arba rekomenduoja pati sistema.
3. Ugdant besimokančiųjų mokėjimo mokytis kompetenciją.

Norint naudoti sukurtą MSMMD modelį būtina taikyti įvairius mokymosi metodus. Siekiant įvertinti taikymo galimybes pradedančiųjų programavimo gebėjimams ugdytis buvo nuspręsta susipažinti su Lietuvos aukštosiose mokyklose dėstomais įvadinio programavimo kursais. Tuo tikslu atliktas žvalgomasis tyrimas, kurio metodika aprašoma kitame skyriuje.

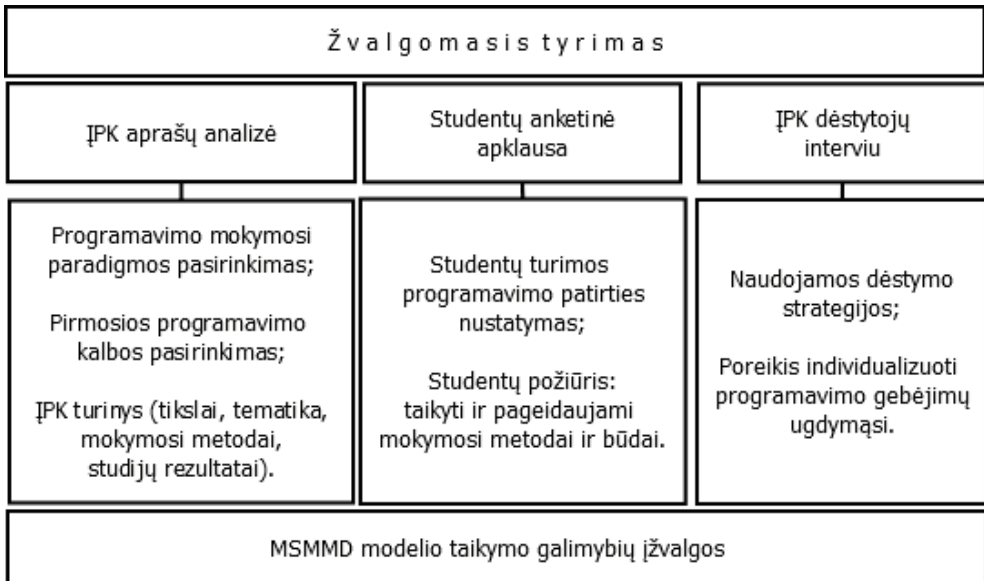
2.3. ŽVALGOMOJO TYRIMO METODIKA

Sprendžiant trečiąją disertacinio tyrimo uždavinį buvo suplanuotas ir atliktas žvalgomasis tyrimas. Žvalgomojo tyrimo tikslas – *numatyti MSMMD modelio taikymo galimybes pradedančiųjų programavimo gebėjimams ugdytis, išanalizuojant, kaip Lietuvos aukštųjų mokyklų (universitetų ir kolegijų) įvadinio programavimo kursuose įgyvendinamas pradedančiųjų programavimo gebėjimų ugdymasis*. Numatyta:

- nustatyti, kuri programavimo paradigma ir kuri programavimo kalba labiausiai tinka pradedantiesiems;
- ištirti įvadinio programavimo kursų tikslus, numatomus studijų rezultatus, tematiką, taikomus studijų metodus;
- atskleisti studentų turimą patirtį pradedant ĮPK studijas;
- apžvelgti naudojamas dėstymo strategijas;
- išryškinti programavimo gebėjimų ugdymosi individualizavimo poreikį.

Kadangi duomenys, gauti iš skirtingų duomenų šaltinių ir surinkti skirtingais duomenų rinkimo metodais, yra tikslesni, išsamesni (Kardelis, 2007), žvalgomasis tyrimas rėmėsi duomenų rinkimo metodų trianguliacija (žr. 9 paveikslą):

- įvadinio programavimo kursų (ĮPK) aprašų analize;
- studentų, išklaususių ĮPK, anketine apklausa;
- interviu su ĮPK dėstančiais dėstytojais.



9 pav. Programavimo gebėjimų ugdymo turinio tyrimo struktūrinė schema

ĮPK įtraukiamas bene į visas su IKT susijusias studijas, tad tyrimo populiacija laikysime pagal visose Lietuvos aukštosiose mokyklose (universitetuose ir kolegijose) vykdomas informatikos, informacijos sistemų, programų sistemų ir informatikos inžinerijos kryptių (IK kryptių) studijų programas studijuojančius studentus.

Atliekant ĮPK aprašų analizę taikytas dokumentų turinio analizės metodas. Analizuota, kokį ĮPK turinį (tikslus, tematiką, studijų metodus, studijų rezultatus) atskleidžia Lietuvos aukštosiose mokyklose dėstomų ĮPK aprašai. Platus kurso studijose taikomų mokymosi metodų spektras užtikrintų galimybę taikyti MSMMD modelį Lietuvos aukštosiose mokyklose siekiant ugdytis programavimo gebėjimus.

Lietuvos universitetuose ir kolegijose (2012–2013 mokslo metų duomenimis) vykdyta 50 IK kryptių studijų programų. Šias studijų programas vykdė 15 kolegijų (27 studijų programos; 54 proc.) ir 8 universitetai (23 studijų programos; 46 proc.). Studijų programų pasiskirstymas pagal studijų kryptis pateiktas 11 lentelėje.

Studijų programų pasiskirstymas pagal studijų kryptis⁴

Mokslų sritis ir kryptis	Studijų programų skaičius					
	Kolegijos		Universitetai		Iš viso	
	Kiekis	Proc.	Kiekis	Proc.	Kiekis	Proc.
Fiziniai mokslai, informatika	5	10	10	20	15	30
Fiziniai mokslai, informacijos sistemos	-	0	3	6	3	6
Fiziniai mokslai, programų sistemos	1	2	2	4	3	6
Technologiniai mokslai, informatikos inžinerija	21	42	8	16	29	58
Iš viso	27	54	23	46	50	100

Dauguma universitetuose vykdomų ĮPK aprašų internete skelbiami viešai, o koleginiuose studijų programose dėstomų dalykų aprašai nėra viešinami. Renkant ĮPK aprašus dokumentų analizei, elektroniniu paštu kreiptasi į aukštųjų mokyklų studijų programas vykdančių katedrų vedėjus, ĮPK dėstančius dėstytojus, paaiškinti atliekamo tyrimo tikslai, pasižadėta neviešinti tyrimui surinktų dokumentų, naudoti tik apdorotus ir apibendrintus duomenis. Dokumentų turinio analizės imtį sudarė 37 dalykų aprašai: 22 (59,5 proc.) universitetuose dėstomų dalykų aprašai ir 15 (40,5 proc.) – kolegijose. Pagal studijų kryptis atrinkti dokumentai pasiskirstė taip: informatikos kryptis – 12 (32,4 proc.); informatikos inžinerijos kryptis – 21 (56,6 proc.); programų sistemų kryptis – 2 (5,4 proc.); informacijos sistemų kryptis – 2 (5,4 proc.).

ĮPK aprašų analizės sritys⁵:

- naudojama programavimo mokymo paradigma ir programavimo kalbos pasirinkimas; analizuoti 37 ĮPK aprašai;
- dalyko tikslai; analizuoti 29 ĮPK aprašai;
- tematika; analizuoti 22 ĮPK aprašai, šiuose kursuose remiamasi procedūrinio programavimo paradigma – tai yra vyraujantis pasirinkimas IK krypties studijų programose (Lietuvoje);
- siekiami studijų dalyko rezultatai; analizuoti 22 ĮPK aprašai;
- studijų metodai; analizuoti 37 ĮPK aprašai.

Studentų anketinė apklausa naudota siekiant išsiaiškinti studentų pasirengimą ĮPK studijoms, šiam dalykui mokytis taikytus ir pageidaujamus mokymosi metodus,

⁴ Šaltinis: *Atvira informavimo konsultavimo ir orientavimo sistema*. Prieiga per internetą: <http://www.aikos.smm.lt/aikos/index.htm>. 2012 m. pab. – 2013 m. pr. duomenys.

⁵ Ne visi analizei pateikti ĮPK aprašai atitiko studijų programų aprašo rengimo metodikoje numatytus reikalavimus. Todėl kai kuriose analizės srityse analizuoti ne visi surinkti dalykų aprašai.

būdas ir studentų požiūrį į šio dalyko studijas. Minėtiems klausimams išsiaiškinti naudotas darbo autorės sudarytas klausimynas. Tyrimo klausimyną (žr. 3 priedą) sudaro instrukcija, kurioje paaiškintas tyrimo tikslas, pateikti 12 diagnostinių ir 5 demografiniai klausimai. Trimis pusiau atvirais klausimais siekta išsiaiškinti studentų turimą programavimo patirtį, įgytą iki studijų aukštojoje mokykloje; dviem pusiau atvirais klausimais – studijų metu taikytus mokymosi metodus ir būdus, likusiais anketos klausimais – studentų požiūrį į aukštojoje mokykloje išklaustą įvadinį programavimo kursą, išžvelgiamus programavimo mokymosi sunkumus ir pageidaujamus mokymosi būdus.

Sudarius anketą ir prieš pradėdant pagrindinę apklausą atliktas bandomasis tyrimas. Jo metu patikslinti klausimai, jų apimtis, pakoreguoti atsakymų variantai.

Siekiant pagrįsti klausimyno vidinį nuoseklumą apskaičiuotas *Cronbach alpha* koeficientas. Pasak J. A. Gliemo ir R. R. Gliem (2003), socialiniuose tyrimuose *Cronbach alpha* koeficiento reikšmės laikomos patikimomis, jeigu jos artimos 0,7, o reikšmės, lygios 0,5 ir mažesnės, yra nepriimtinos (Gliem, Gliem, 2003). Klausimyną sudarė dvi dalys, siekta: 1) nustatyti respondentų turimą programavimo patirtį; 2) identifikuoti studijų metu taikytus mokymosi metodus ir būdus. *Cronbach alpha* koeficiento reikšmės apskaičiuotos atskiroms klausimyno dalims: 1) $\alpha = 0,68$; 2) $\alpha = 0,60$. Remiantis gautais rezultatais galima teigti, kad respondentų programavimo iki studijų patirtis atskleista, tačiau kita tirta problema, susijusi su studijų metu taikytais mokymosi metodais ir būdais, išaiškinta nepakankamai gerai. Siekiant geriau atskleisti studijų metu taikytus mokymosi metodus ir būdus suplanuotas ir atliktas ĮPK dėstančių dėstytojų interviu.

Anketinio tyrimo imtis sudaryta taip, kad jos ypatybės atspindėtų generalinės visumos ypatybes, t. y. ji turi būti reprezentatyvi. Socialiniams tyrimams rekomenduojama imties dydį nustatyti atsižvelgiant į 5 procentų paklaidą (Kardelis, 2007). Tad imties dydis nustatytas pagal formulę (Паниотто, Максименко, 2003):

$$n = \frac{1}{\Delta^2 + \frac{1}{N}},$$

čia n – imties dydis; Δ – paklaidos dydis; N – generalinės aibės dydis. Skaičiuojant imtį pasirinktas 0,05 paklaidos dydis.

Priimtųjų į IK kryptių programas 2011–2013 metais⁶ skaičiai

Koleginės studijos	2011 m.	2012 m.	2013 m.	Iš viso	Proc.
K1: Informatikos	214	213	205	632	11,0
K2: Informatikos inžinerijos	601	673	743	2017	34,0
K3: Programų sistemų	23	28	23	74	1,0
Iš viso kolegines studijas	838	914	971	2723	47,0
Universitetinės studijos, pirmoji pakopa	2011 m.	2012 m.	2013 m.	Iš viso	Proc.
K1: Informatikos	323	302	466	1091	19,0
K2: Informatikos inžinerijos	383	381	376	1140	19,0
K3: Programų sistemų	187	203	278	668	42,0
K4: Informacijos sistemų	85	64	77	226	4,0
Iš viso universitetinės studijos, pirmoji pakopa	978	950	1197	3125	53,0
Iš viso	1816	1864	2168	5848	100,0

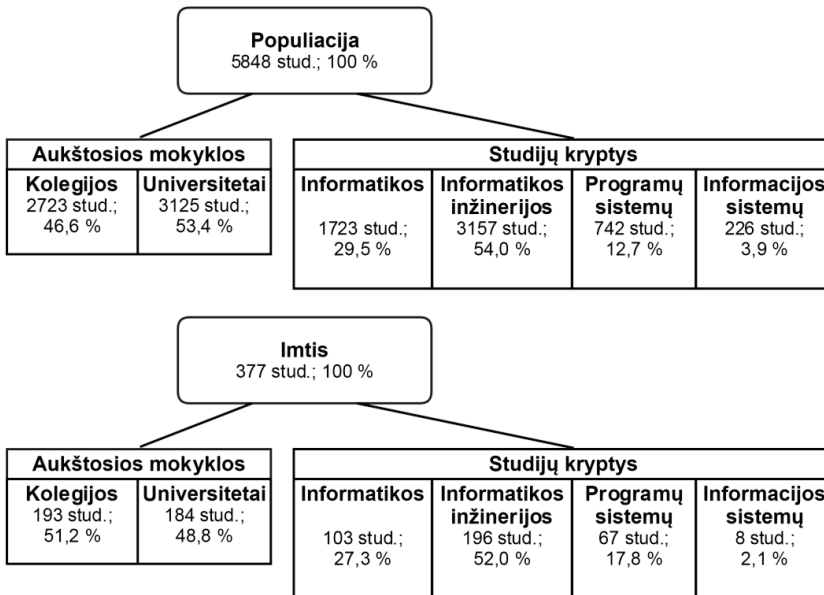
Anketinio tyrimo generaline aibe laikysime IK kryptių studijų programų pirmojo arba antrojo kurso studentus, kurie yra išklause įvadinio programavimo kursą. Kadangi nuo 2012 metų statistinė informacija apie aukštąjį mokslą rengiama remiantis Švietimo informacinių technologijų centro pateiktais statistiniais duomenimis, siekiant išsiaiškinti generalinės aibės dydį (žr. 12 lentelę), kreiptasi į Lietuvos statistikos departamentą ir Švietimo informacinių technologijų centrą.

Per 2011–2013 metus IK kryptių studijų programas pasirinko 5 848 studentai. Apskaičiuotas bendras anketinio tyrimo imties dydis:

$$n = \frac{1}{0,05^2 + \frac{1}{5848}} = 374 \text{ resp.}$$

Tyrimo imčiai sudaryti taikytas netikimybinis atrankos būdas. Jį taikant nenumatoma laikytis atsitiktinumo principo, t. y. čia tiriamos visumos narių ar elementų patekimo į imtį tikimybė nėra vienoda. Netikimybinės atrankos būdu sudaryta imtis gali būti tiek reprezentatyvi, tiek nerepresentatyvi – tai priklauso nuo tyrimo tikslų ir technikų imčiai sudaryti. Imties respondentams atrinkti taikyta kvotinė atranka. Siekta užtikrinti, kad atrinktą visumą sudarytų toks pat skirtingų charakteristikų tiriamųjų santykis, kaip ir tiriamoje visumoje (Kardelis, 2007). Tyrimo imties dalyvių pasiskirstymas pagal įvestas kvotas pateiktas 10 paveiksle.

⁶ Šaltiniai: 2011 m. LR statistikos departamentas; 2012–2013 m. Švietimo informacinių technologijų centras.



10 pav. Anketinio tyrimo populiacija ir imtis

Anketinei apklausai atlikti pasirinktas internetinės apklausos įrankis *mano.apklausa.lt*. Internetinės apklausos socialinių mokslų tyrimuose yra gana naujas, tačiau sparčiai populiarėjantis duomenų rinkimo metodas, lemiantis santykinai žemas laiko ir finansines sąnaudas, gerą tiriamųjų pasiekiamumą ir operatyvų informacijos gavimą. Klausimyno internetinis adresas respondentams perduotas per ĮPK dėstytojus, o vienu atveju – per studentų atstovybę. Apklausos dalyviai informuoti, kad vykdoma apklausa yra anoniminė, o duomenys bus naudojami tik mokslo tikslais.

Gauti apklausos duomenys buvo apdorojami ir analizuojami naudojant statistinį duomenų analizės paketą *SPSS 16* ir programinį paketą *Microsoft Excel 2007*. Taikyta aprašomoji statistika (absoliučių ir procentinių dažnių skaičiavimai). *Chi* kvadrato kriterijus pasitelktas siekiant nustatyti, ar egzistuoja statistiškai reikšmingas skirtumas tarp duomenų grupių (Čekanavičius, Murauskas, 2000).

Įvadinio programavimo kursą dėstančių dėstytojų interviu. Tyrimo duomenims rinkti pasirinktas metodas – pusiau struktūruotas interviu, kuriam iš anksto parengtas aptariamų klausimų sąrašas (žr. 4 priedą) ir jų pateikimo seka. Tačiau tyrimo metu kai kuriais atvejais buvo užduodami papildomi klausimai. Taip siekta gauti daugiau informacijos, kai tiriamasis nevisiškai atsakė į pateiktą klausimą. Interviu dalyviai –

Lietuvos kolegijų ir universitetų dėstytojai, dėstantys įvadinį programavimo kursą. Pokalbio metu siekta išsiaiškinti dėstytojų naudojamas šio dalyko mokymo strategijas. Interviu įrašinėti gavus kiekvieno informanto žodinį sutikimą, kuris taip pat įrašytas kiekvieno interviu pradžioje. Įrašo pradžioje nurodytas informanto vardas ir pavardė bei tyrėjos pažadas publikuoti tyrimo duomenis tik apdorotus, t. y. anoniminius. Tyrime sutiko dalyvauti 5 dėstytojai, dėstantys ĮPK universitetinėse ir / arba kolegijinėse studijų programose. Demografiniai informantų duomenys pateikti 13 lentelėje.

13 lentelė

Demografinės informantų charakteristikos

Kodas	Lytis	Profesinė patirtis		Mokymo įstaiga, miestas	Pokalbio trukmė, min.
		Bendra	Dėstymo ĮPK aukštojoje mokykloje		
In-1	Moteris	16	3	Universitetas, Kaunas	58
In-2	Vyras	7	6	Kolegija, Vilnius	53
In-3	Moteris	19	12	Universitetas, kolegija, Vilnius	43
In-4	Vyras	5	4	Kolegija, Kaunas	23
In-5	Vyras	15	7	Universitetas, Vilnius	54
Bendra pokalbių trukmė					231

Tyrimas buvo atliekamas 2013 m. gruodžio mėn. – 2014 m. vasario mėn. Į tyrimą įtraukti informantai, sutikę dalyvauti; dėl tikslaus interviu laiko su jais susitarta iš anksto. Pokalbiai su informantais iš Vilniaus vyko Lietuvos edukologijos universiteto patalpose, o pokalbiams su informantais iš Kauno organizuoti ir įrašyti pasitelktas interneto telefonijos tinklas bei kompiuterinės programos *Skype* ir *Power Gramo*. Kiekvieno pokalbio trukmė buvo skirtinga, maždaug nuo 23 iki 58 minučių. Bendra pokalbių trukmė siekia 231 minutę. Pokalbiai transkribuoti, o duomenys tvarkyti naudojantis transkribuotu tekstu. Prasminiu vienetu laikoma frazė, atskleidžianti asmens požiūrį tam tikru klausimu. Apdorojant duomenis visi informantai užkoduoti, duomenų analizei atlikti naudotas kokybinių duomenų analizės paketas *QSR Nvivo 8* – patogi priemonė atliekant tam tikras operacijas, tačiau tik pats tyrėjas yra atsakingas už mąstymą ir interpretavimą. Tokią programinę įrangą lengviausia suvokti kaip duomenų bazę, kurioje saugomi prasminiai vienetai (informantų pasisakymai), tyrėjo sudarytos kategorijos ir jų tarpusavio ryšiai, vėliau nesudėtingai surandami. Programinėje įrangoje šios kategorijos vadinamos kodais (anglų k. *nodes*), o tarpusavio ryšiai – sąsajomis (anglų k. *links*),

kurios sukuriamos analizuojant duomenis. Kai duomenys užkoduoti, norint juos ištyrinėti ar nustatyti temas ir struktūras galima pasinaudoti paieškos algoritmais (Morkevičius, Telešienė, Žvaliauskas, 2008).

2.4. VEIKLOS TYRIMO METODIKA

Sudarytam MSMMD modeliui tirti, kai siekiama išryškinti mokymosi stilių ir mokymosi metodų dermės veiksnį, pasirinktas veiklos tyrimas. S. Kemmis ir R. McTaggartas (2005) išskiria keletą veiklos tyrimų tipų:

- klasės veiklos tyrimą paprastai atlieka mokytojai (kartais kartu su profesionaliais tyrėjais) siekdami patobulinti pedagoginės veiklos praktikas mokykloje;
- mokymasis per veiklą kaip tyrimo strategija dažniausiai taikomas vadyboje sprendžiant organizacijų problemas. Esminė idėja – suburti drauge darbuotojus ir jų vadovybę tam, kad organizacijos nariai mokytųsi vieni iš kitų, kad geriau suvoktų savo vaidmenį organizacijoje, suprastų, ko organizacija siekia, ir kartu imtųsi spręsti iškilusias problemas;
- Industrinės veiklos tyrimai dažniausiai atliekami organizacijų psichologijos ir organizacijų vystymo tyrimuose. Tokiame tyrime svarbiausia – konsultavimas, t. y. profesionalus tyrėjas ar tos srities žinovas teikia ekspertines konsultacijas organizacijos nariams (Kemmis, McTaggart, 2005; Morkevičius, Telešienė, Žvaliauskas, 2008).

Remiantis šia veiklos tyrimų klasifikacija, disertacinis veiklos tyrimas priskirtinas klasės veiklos tyrimams (kitai edukacinės veiklos), nes veikla orientuojama į ugdymosi proceso turinio kaitą ir tobulinimą. Edukacinis veiklos tyrimas mokslinėje literatūroje apibrėžiamas kaip sistemingas tyrinėjimas siekiant keisti esamą padėtį, ieškoti naujų veiklos būdų, visiems ugdymosi proceso dalyviams užtikrinančių teigiamų pokyčių (Adomaitienė, Teresevičienė, Zubrickienė, 2007; Juodaitytė, 2012; Lichtman, 2013). Edukacinis veiklos tyrimas disertaciniame darbe panaudotas sudarytam MSMMD modeliui verifikuoti ir tobulinti. Veiklos tyrimo metodas tinkamas disertacijos idėjai įgyvendinti, nes atliekant veiklos tyrimą tikslo siekiama nevisiškai įsitikinus savo teisumu (Merfeldaitė, 2007).

Veiklos tyrimas grindžiamas šiais principais:

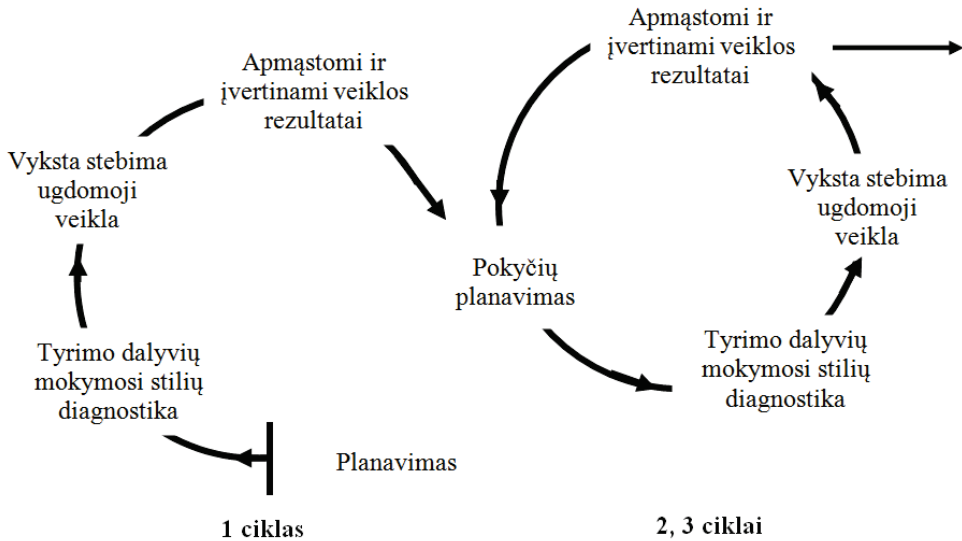
- *ugdomosios veiklos tikslingumo* – ugdomoji veikla atliekama sąmoningai, tikslingai;
- *bendradarbiavimo* – tyrėjas ir tiriamieji veiklos metu aktyviai bendradarbiauja;

- *kritinio veiklos apmąstymo* – ugdomoji veikla kritiškai apmąstoma ir įvertinama. Veiklos tyrimą atliekantis tyrėjas ar praktikas yra ne vien kritiškas, bet ir savikritiškas;
- *kaitos* – veikla yra lanksti ir atvira pokyčiams;
- *cikliškumo* – atliekant veiklos tyrimą cikliškai kartojami keturi pagrindiniai etapai: planavimas, ugdomoji veikla, stebėjimas, apmąstymas (refleksija);
- *teorijos, praktikos vienovės* – teorija teikia informacijos praktikai, praktika tobulina teoriją (Bitinas, 2006; Merfeldaitė, 2007; Railienė, 2010). Tiriama konkreti praktinė, o ne abstrakti veikla.

Atliekant veiklos tyrimą sprendžiamas disertacinio tyrimo uždavinys: atskleisti mokymosi stilių ir mokymosi metodų dermės raišką, ugdantis programavimo gebėjimus aukštojoje mokykloje.

Veiklos tyrimas kartotas tris kartus su trimis skirtingomis tiriamųjų grupėmis:

- 2012 m. rugsėjo mėn. – 2012 m. gruodžio mėn.: veiklos tyrimo I ciklas.
- 2013 vasario mėn. – 2013 gegužės mėn.: veiklos tyrimo II ciklas.
- 2013 rugsėjo mėn. – 2013 gruodžio mėn.: veiklos tyrimo III ciklas.



11 pav. Atlikto veiklos tyrimo schema

Kiekvienam veiklos tyrimo ciklui būdinga analogiška struktūra (žr. 11 paveikslą):

1. **Planavimas** – pasiruošimas veiklai. Pradedant veiklos tyrimo pirmąjį ciklą parengta ir nepriklausomo eksperto įvertinta⁷ įvadinio programavimo kurso programa, į lietuvių kalbą išverstas⁸ ir apribotas P. Honey ir A. Mumfordo mokymosi stilių klausimynas. Ugdomųjų situacijų scenarijai parengti remiantis sudarytu MSMMD modeliu. Kai veiklos tyrimas kartotas (antrasis ir trečiasis ciklai), šiame etape, remiantis veiklos proceso apmąstymo ir studentų apklausų rezultatais, suplanuojami pokyčiai, numatoma, kaip patobulinti ugdomąją veiklą.
2. **Mokymosi stiliaus diagnostika** – naudojamas pasirinktos P. Honey ir A. Mumfordo mokymosi stilių tipologijos klausimynas.
3. **Vykdyto etapas** – organizuojama stebima ugdomoji veikla. Įvadinio programavimo kurso dėstymo metu įgyvendinami planavimo etape parengti ugdomosios veiklos scenarijai. Šiame etape kaupiami įvairūs kokybiniai tyrimo duomenys, studentų darbai, studentų apklausos (ugdomųjų veiklų vertinimai), stebėjimų užrašai.
4. **Veiklos apmąstymo ir įvertinimo etapas** – veiklos procesas ir pasekmės apmąstomi ir įvertinami. Šis etapas skirtas veiklos tyrimo rezultatams apibendrinti, įvadinio programavimo kurso programai ir MSMMD modeliui tobulinti, remiantis grįžtama informacija ir stebėjimų rezultatais.

Duomenys renkami taikant kokybinių tyrimų metodus, atskleidžiančius veiklos tyrimo dalyvių nuomones ir požiūrius, veiklos niuansus ir geresnį veiklos sąlygų supratimą (Morkevičius, Telešienė, Žvaliauskas, 2008). Dėl dalyvavimo principo šiame tyrime rečiau taikomi kiekybiniai metodai, nes tyrėjas taptų nešališkas, atitolęs stebėtojas. Veiklos tyrimo metu taikyti tokie duomenų rinkimo metodai:

- *tyrėjos stebėjimų įrašai, asmeninės refleksijos tekstai* – tai tyrėjos sukurti rašytiniai dokumentai – struktūruotos ir nestruktūruotos refleksijos raštu;
- *veiklos tyrimo dalyvių apklausos* – tai trumpos apklausos, kurių metu dalyviai vertino taikytus mokymosi metodus, pildydami vertinimo formas iškart po atliktos veiklos. Vertinimo formų pavyzdžiai pateikiami 8 priede;
- *veiklos tyrimo dalyvių interviu* – tai pusiau struktūruotas interviu, kuriam iš anksto parengtas aptariamų klausimų sąrašas (žr. 5 priedą). Iš pat pradžių, planuojant veiklos tyrimą, nebuvo numatyta imti tyrimo dalyvių interviu.

⁷ Vertinimas atliktas įgyvendinant projekto „Vilniaus verslo kolegijos studijų programų tarptautiškumo didinimas“ (Nr. VP1-2.2-ŠMM-07-K-02-04) studijų programos „Programavimas ir internetinės technologijos“ atnaujintų dalykų (modulių) ekspertinį vertinimą.

⁸ Vertimą atliko dvikalbių ekspertų grupė.

Tačiau išanalizavus ir įvertinus pirmojo veiklos tyrimo ciklo pabaigoje sukauptus duomenis, nutarta patobulinti veiklos tyrimo metodiką, įtraukti ir kitą duomenų rinkimo metodą – veiklos tyrimo dalyvių interviu. Interviu organizuotas veiklos ciklo pabaigoje ir taikytas baigiant antrąjį ir trečiąjį veiklos tyrimo ciklus. Interviu sudalyvavo 14 veiklos tyrimo dalyvių.

Metodologinės trianguliacijos taikymas veiklos tyrime ir duomenų rinkimo kar-tojimas kitomis sąlygomis (duomenys renkami trijų ciklų metu) laikytini esminiais tyrimo patikimumą pagrindžiančiais elementais.

Veiklos tyrimo dalyviai. Veiklos tyrimo dalyvių grupėms priskirtini disertantė ir studentai, įvadinio programavimo kurso klausytojai. Pagrindinis veiklos tyrimo dalyvių (studentų) atrankos kriterijus – savanoriškumo principas.

Pradedant kiekvieną veiklos tyrimo ciklą pagal adaptuotą P. Honey ir A. Mumfordo mokymosi stilių tipologijos klausimyną nustatytas tyrimo dalyvių mokymosi stilius. Klausimyną sudaro 80 besimokančiojo elgseną vertinančių teiginių, su kuriais klausimyną pildantis asmuo gali sutikti arba nesutikti. Klausimyną sudaro 4 grupės po 20 teiginių, apibūdinančių kiekvieną mokymosi stilių (aktyvistas, mąstytojas, teoretikas, pragmatikas). Kad būtų nustatytas vyraujantis mokymosi stilius, skaičiuojama kiekvienos patvirtintų teiginių grupės suma. Tačiau ne visuomet mokymosi stilius yra aiškus. Gali būti, kad keliose teiginių grupėse surenkamas vienodas balų skaičius, tokiu atveju besimokančiajam būdingi keli mokymosi stiliai; arba visose teiginių grupėse surenkama labai mažai balų, tai leidžia teigti, kad besimokančiojo mokymosi stilius yra neaiškus. P. Honey ir A. Mumfordo teigimu, surinkti balai gali būti ranguojami: išskiriami labai stipriai, stipriai, vidutiniškai, silpnai ir labai silpnai išreikštus mokymosi stilių lygiai (žr. 14 lentelę).

14 lentelė

**P. Honey ir A. Mumfordo pateiktas mokymosi stiliaus raiškos stiprumas
(Honey, Mumford, 2000)**

Mokymosi stiliaus raiškos stiprumas	Aktyvistas	Mąstytojas	Teoretikas	Pragmatikas
Labai stiprus	13–20	18–20	16–20	17–20
Stiprus	11–12	15–17	14–15	15–16
Vidutinis	7–10	12–14	11–13	12–14
Silpnas	4–6	9–11	8–10	9–11
Labai silpnas	0–3	0–8	0–7	0–8

Nustačius veiklos tyrimo dalyvių mokymosi stilius, atsižvelgta ir į jų išreikštumą (žr. 15 lentelę). Pastebėta, kad tarp tiriamųjų vyrauja mąstytojo mokymosi stilius. Viso tyrimo metu labai stipriai ir stipriai išreikštą mąstytojo mokymosi stilių turėjo daugiau kaip pusė tiriamųjų. Antras pagal populiarumą – aktyvisto mokymosi stilius, o net 5 veiklos tyrimo dalyviai turėjo stipriai išreikštą aktyvisto ir mąstytojo mokymosi stilių derinį. Vienas iš tiriamųjų turėjo net tris labai stipriai išreikštus mokymosi stilius: aktyvisto, mąstytojo, teoretiko. Toks mokymosi stilių derinys leidžia rinktis labai platų spektrą tinkamų mokymosi veiklų, o pats besimokantysis paprastai nepatiria jokių mokymosi sunkumų. Deja, tyrimo metu pasitaikė labai mažai tiriamųjų, turinčių tik teoretiko arba tik pragmatiko mokymosi stilių.

15 lentelė

Tiriamųjų, dalyvavusių veiklos tyrime (N = 39), pasiskirstymas pagal jų mokymosi stilius

Mokymosi stiliaus raiškos stiprumas	A – aktyvistas	M – mąstytojas	T – teoretikas	P – pragmatikas
4 – labai stiprus	8 20,51 %	5 12,82 %	5 12,82 %	1 2,56 %
3 – stiprus	6 15,38 %	16 41,03 %	4 10,26 %	5 12,82 %
2 – vidutinis	14 35,90 %	10 25,64 %	20 51,28 %	12 30,77 %
1 – silpnas	9 23,08 %	5 12,82 %	7 17,95 %	13 33,33 %
0 – labai silpnas	2 5,13 %	3 7,69 %	3 7,69 %	8 20,51 %
Reikšmių vidurkis	8,87	14,21	11,92	11,26
Standartinis nuokrypis	3,67	3,28	2,79	3,20

Planuojant atliekamo tyrimo veiklas, tiriamieji, turintys keletą vienodai išreikštų mokymosi stilių, gali būti priskiriami tiek vieno, tiek kito stiliaus atstovų grupei. Tačiau priskyrus prie kurios nors vienos grupės tyrimo rezultatai gali būti netikslūs, kai reiks apibendrinti duomenis. Tad nuspręsta, kad aptariant tyrimo duomenis bus sudaryta atskira tokių tiriamųjų grupė. Vykdamas kiekvieną veiklos tyrimo ciklą, atrinkti įvairių mokymosi stilių tiriamieji. Stebėti pasirinkta po 7 dalyvius, kurių mokymosi stiliai ir jų priskirimas prie grupių pateikti 16 lentelėje. Suformuotos šešios mokymosi stilių grupės (0 – mokymosi stilius neišreikštas; 1 – stipriai išreikštas tik aktyvisto mokymosi stilius; 2 – stipriai išreikštas tik mąstytojo mokymosi stilius; 3 – stipriai išreikštas tik teoretiko mokymosi stilius; 4 – stipriai išreikštas tik pragmatiko mokymosi stilius; 5 – stipriai išreikšta keletas mokymosi stilių).

Stebėtų informantų mokymosi stilių išreikštumas ir priskyrimas mokymosi stilių grupėms

Veiklos tyrimo ciklai	Informanto kodas	Mokymosi stiliaus išreikštumas				Mokymosi stilių grupės
		A	M	T	P	
1 ciklas	P1	2	3	3	2	5 – MT
	P2	3	4	2	1	5 – MA
	P3	4	0	2	2	1 – A
	P4	2	2	2	3	4 – P
	P5	1	3	2	2	2 – M
	P6	1	2	2	3	4 – P
	P7	2	2	2	1	0 – neišreikštas
2 ciklas	L1	4	2	2	3	5 – AP
	L2	3	2	1	1	1 – A
	L3	0	2	0	1	0 – neišreikštas
	L4	2	0	0	0	0 – neišreikštas
	L6	1	1	4	2	3 – T
	L8	2	3	2	2	2 – M
	L9	1	3	2	1	2 – M
3 ciklas	V4	0	3	2	2	2 – M
	V7	2	3	4	4	5 – TPM
	V8	2	2	2	0	0 – neišreikštas
	V9	4	1	2	2	1 – A
	V10	2	3	1	1	2 – M
	V18	1	3	4	0	5 – TM
	V20	2	2	3	1	3 – T

Mokymosi stilių klausimyno (LSQ) vertimas ir aprobavimas. Klausimyną į lietuvių kalbą vertė du laisvai samdomi vertėjai. Rengiant lietuviškąją klausimyno versiją laikytasi lingvistinio lygiavertiškumo principo, mat verčiant teiginius į lietuvių kalbą daug svarbiau išlaikyti teiginio reikšmę, nei išversti pavienius žodžius. Vertimo procesas turi sudaryti sąlygas tam, kad originaliosios ir adaptuotosios versijų klausimai reikštų tuos pačius dalykus skirtingiems žmonėms. Parengtą lietuviškojo klausimyno versiją redagavo lietuvių kalbos ir stilistikos specialistas, o vertimus peržiūrėjo dvikalbių ekspertų grupė (3 ekspertai). Posėdžio metu sulyginami dviejų laisvai samdomų vertėjų vertimų tekstai. Atliktas ir atgalinis vertimas; šį palyginus su originalu esminių skirtumų nenustatyta.

Klausimyno skalės vidiniam suderinamumui (anglų k. *scale internal consistency*) įvertinti dažniausiai skaičiuojamas *Cronbach alpha* (α) koeficientas. *Cronbach alpha* koeficientas remiasi atskirų klausimų, sudarančių klausimyną, koreliacija ir įvertina, ar visi skalės klausimai tinkamai atspindi tiriamąjį dydį. *Cronbach alpha* koeficiento skaičiavimas paremtas dispersine analize. Jeigu atskirų klausimų dispersijų suma yra artima visos skalės dispersijai, vadinasi, atskiri klausimai tarpusavyje nekoreliuoja, t. y. jie neatspindi to paties dalyko (Yaffee, 2003, cituojama Pukėnas, 2009):

$$\alpha = \frac{k}{k-1} \left(1 - \frac{\sum_{i=1}^k S_i^2}{S^2} \right),$$

čia k – klausimyno diagnostinių kintamųjų skaičius; S_i^2 – i -ojo kintamojo empirinė dispersija; S^2 – viso klausimyno empirinė dispersija. Apskaičiuota lietuviškos LSQ klausimyno versijos *Cronbach alpha* koeficiento reikšmė lygi 0,82. Atskirų klausimyno skalių (aktyvisto, mąstytojo, teoretiko, pragmatiko) *Cronbach alpha* koeficientų reikšmės pateiktos 17 lentelėje. Tais atvejais, kai klausimyno įverčiai skalėje yra dichotominiai (atsakymai: „sutinku“ / „nesutinku“), tikrinant vidinį suderinamumą skaičiuojamas *Kuder-Richardson KR20* koeficientas:

$$KR20 = \frac{k}{k-1} \left(1 - \frac{\sum_{i=1}^k p_i \times q_i}{S^2} \right),$$

čia k – klausimyno diagnostinių kintamųjų skaičius; p_i – respondentų, i teiginiui pasirinkusių atsakymą „sutinku“, skaičius; q_i – respondentų, i teiginiui pasirinkusių atsakymą „nesutinku“, skaičius; S^2 – viso klausimyno empirinė dispersija. Apskaičiuotos viso LSQ klausimyno ir jo atskirų skalių *KR20* koeficiento reikšmės parodė tą patį vidinį suderinamumą.

Remiantis gautais klausimyno skalės vidinio suderinamumo vertinimo rezultatais galima teigti, kad lietuviškosios LSQ klausimyno versijos struktūra pakankamai patikima, o gautos koeficientų reikšmės artimos kitų tyrėjų (DeCiantis, Kirton, 1996; Van Zwanenberg ir kt., 2000; Pickworth ir kt., 2000) apskaičiuotosioms.

**Atskirų P. Honey ir A. Mumfordo klausimyno dalių vidinis suderinamumas
(N = 74)**

Klausimyno dalys	Vidurkis (skalė: 1–20)	Standartinis nuokrypis	Cronbach α , KR20
Aktyvistas	8,8	4,1	0,77
Mąstytojas	14,2	3,2	0,66
Teoretikas	12,4	3,3	0,63
Pragmatikas	12,0	3,5	0,69

Veiklos tyrimo duomenų analizė. Atliekant kokybinių duomenų analizę taikytas kokybinės turinio analizės metodas. J. Creswellas (2007) teigia, kad pačia bendriausia prasme kokybinę duomenų analizę sudaro trys pagrindiniai procesai: 1) duomenų parengimas analizei, 2) jų redukavimas į temas atliekant kodavimą ir 3) rezultatų pateikimas grafikuose ar lentelėse (Creswell, 2007, cituojama Morkevičius ir kt., 2008). Kokybinių duomenų analizę sudarė keletas žingsnių:

1 žingsnis: duomenų parengimas analizei. Interviu metu sukaupti duomenys sugrupuoti į atskiras grupes pagal aptartus mokymosi metodus.

2 žingsnis: analizės vieneto pasirinkimas – vienetu pasirinkta frazė, atskleidžianti tiriamojo požiūrį tam tikru klausimu.

3 žingsnis: subkategorijų ir kodavimo schemos sudarymas. Analizuojant prastinius vienetus sudarytos subkategorijos, kurios gautos euristiniu būdu: skaitant tiriamąjį tekstą abstrahuojama išsakyta informacija. Išskirtos subkategorijos buvo suskirstytos į tam tikras grupes. Tai stambiausia kodavimo kategorija. Atskleisti mokymosi stiliaus ir mokymosi metodo dermę analizuojant vien tik interviu duomenis buvo gana sudėtinga, būtent todėl informantų pasisakymai buvo papildomi ir stebėjimų metu sukaupta informacija.

4 žingsnis: koduotos medžiagos apibendrinimas. Koduotų tekstų turiniai pateikiami lentelėse.

Veiklos tyrimo rezultatų analizės pagrindu patobulintas mokymosi metodų ir mokymosi stilių dermės modelis.

2.5. TYRIMO ETIKA

Svarbiausias mokslinių tyrimų uždavinys – pažinti pasaulį, ieškoti naujų tiesų apie reiškinius, procesus ir objektus. Vis dėlto labai svarbu nepamiršti, kad mokslininko ar jų grupės atliekami moksliniai tyrimai turi remtis ir mokslinės etikos principais.

Vienas iš atlikto tyrimo etikos klausimų – ar tiriamieji, apie tyrimą gavę išsamios informacijos, sutiks dalyvauti tyrime. Mokslinio tyrimo dalyviai – aukštųjų mokyklų atstovai (katedrų vedėjai, direktoriai, ĮPK dėstytojai) ir studentai. Su aukštųjų mokyklų atstovais susisiekti renkant duomenis dokumentų analizei. Į aukštųjų mokyklų atsakingų padalinių vadovus ir / ar įvadinio programavimo kursų dėstytojus kreiptasi elektroniniu paštu (dalykiniam susirašinėjimui naudota institucinio elektroninio pašto dėžutė (...@leu.lt)), pristatyti atliekamo mokslinio tyrimo tikslai ir tyrėjos asmenybė. Aukštųjų mokyklų atstovams pažadėta išlaikyti turinio analizei pateiktų dokumentų konfidencialumą, tyrimo duomenis skelbti tik apibendrintus ir užkoduotus. Pažadėta grįžtamoji informacija – viešas pranešimas, atlikto tyrimo rezultatų pristatymas Lietuvos kompiuterininkų sąjungos organizuojamoje konferencijoje „Kompiuterininkų dienos – 2013“.

Kitas etikos klausimas susijęs su aukštųjų mokyklų, kurių studentai dalyvavo veiklos tyrime, informuotumu ir leidimų atlikti tyrimus gavimu. Apie vykdomą mokslinį tyrimą informuoti atsakingieji asmenys. Institucijų atstovai pritarė ir palaikė tyrėją, atliekančią mokslinį tyrimą.

Tiriamieji asmenys gali būti suskirstyti į tris grupes:

1. Anketinio tyrimo dalyviai – tai įvairių aukštųjų mokyklų pirmojo ir antrojo kursų studentai, išklausę įvadinio programavimo kursą.
2. Interviu dalyviai – keleto aukštųjų mokyklų įvadinio programavimo kurso dėstytojai, davę tyrėjai žodinį interviu.
3. Veiklos tyrimo dalyviai – tai informatikos studijų krypties specialybių studentai.

Anketinėje apklausoje dalyvavusiems studentams klausimyno įžangoje pristatytas tik anketinio tyrimo tikslas, o jo svarba atliekamam moksliniam tyrimui nedetaliazuojama. Kur kas išsamiau paaiškinta mokslinio tyrimo problematika interviu davusiems informantams – dėstytojams. Šiems pristatytas atliekamo tyrimo tikslas ir uždaviniai, gautas jų sutikimas dalyvauti tyrime, paaiškinta, kur ir kaip bus panaudota gauta informacija. Veiklos tyrimo dalyviams studijų semestro pradžioje, pristatant studijuojamą dalyką, tuo pat metu pristatytas ir tyrėjos-dėstytojos atliekamas mokslinis tyrimas. Studentams pasiūlyta prisidėti prie atliekamo mokslinio tyrimo

ir suteikti tyrimui reikiamą informaciją. Tyrimo dalyviams paaiškinta galima nauda, t. y. jų indėlis į tyrimo srities supratimą. Visi dalyviai buvo informuoti ir žinojo, kad jų dalyvavimas tyrime yra savanoriškas, be to, turėjo galimybę atsisakyti dalyvauti tyrime net ir jam prasidėjus.

Dalyviams – studentams ir dėstytojams – užtikrintas anonimiškumas. Tyrimo metu kaupti duomenys užkoduoti, pateikiama informacija neleidžia nustatyti tiriamųjų tapatybės. Anonimiškumą garantuoja stebėjimų įrašai ir anketos be vardų ar kitų asmenis leidžiančių identifikuoti duomenų. Veiklos tyrimo dalyvių anonimiškumas nėra absoliutus, bet „tyrime domimasi besimokančiųjų elgsena apskritai, o ne atskirų individų elgesiu, todėl pats tyrimo beasmeniškumas turi didelį etinį privalumą, nes padeda išvengti nepageidaujamų pasekmių ir privatumo pažeidimo“ (Kardelis, 2007).

Apibendrinant galima teigti, kad tiriamųjų teisėms skirta gana daug dėmesio. Bet kokie prieštaravimai buvo gerbiami: pavyzdžiui, ne visi veiklos tyrime dalyvavę tiriamieji sutiko duoti baigiamąjį interviu, kurio metu buvo aptariamoms jų mokymosi sėkmių ir nesėkmių priežastys.

2.6. TYRIMO APRIBOJIMAI

Dokumentų turinio analizės apribojimai tiesiogiai susiję su tyrime naudotų įvadinio programavimo kursų aprašų kokybe. Atliekant dokumentų turinio analizę nebuvo keltas uždavinys vertinti kiekvieno studijų dalyko aprašo kokybę. Disertantės nuomone, visiškai patikimais laikytini „demografiniai“ dokumentų duomenys: studijų dalyko pavadinimas, skiriami kreditai, dalyko vieta studijų programoje ir deklaruojamos programavimo mokymosi paradigma ir pirmoji programavimo kalba. Pagal IPK aprašų tematiką galima išskirti dvi kategorijas: 1) išsamūs aprašai ir 2) apibendrinti aprašai. Išsamiais laikytini dalykų aprašai, kuriuose pateikiamos pagrindinės temos, detalizuojamos potėmėmis. Tokių aprašų tematiką galima laikyti visiškai atskleista. Tačiau apibendrintuose dalykų aprašuose nurodomos tik pagrindinės temos ir praleidžiamos smulkesnės temų potėmės. Remiantis tokiais aprašais negalima įvertinti, ar tema yra išsamiai nagrinėjama. Būtent dėl šios priežasties kurso tematika nebuvo visiškai atskleista.

Vis dėlto dėl didelės mokslo ir technikos pažangos sparčiai kinta ir reikalavimai normatyviniams aukštojo mokslo dokumentams bei teorinėms nuostatoms, kuriomis būtina remtis rengiant studijų dokumentus. Studijų kryptių aprašų rengimo

metodikose⁹ pateikiama ir rekomendacijų, kaip formuluoti studijų dalyko tikslus ir studijų dalyko rezultatus. Be to, per pastaruosius penkerius metus bene visos Lietuvos aukštosios mokyklos susikūrė vidinę studijų kokybės vadybos sistemą, tad šie dalykai turėtų pagerinti ir studijų dokumentų kokybę.

Anketinio tyrimo apribojimai susiję su imčiai formuoti taikytu netikimybinium kvotinės atrankos būdu. Vis dėlto sudaryta imtis reprezentatyvi aukštųjų mokyklų ir studijų programų kryptį atžvilgiu (paklaida neviršija 5 proc.).

Kokybinių duomenų analizės apribojimai. Kokybinės turinio analizės rezultatus visuomet įtakos turi tam tikros dalies tyrėjo subjektyvios interpretacijos, nes analizuojami tekstai yra įvairiaprasmiai (Bitinas, Rupšienė, Žydžiūnaitė, 2008).

Kaip apribojimas vertinamas ir nepavykęs informantų atsakymas į klausimą dėl nežinojimo, užmiršimo ar nesugebėjimo išreikšti mintį. Visais atvejais tai gali būti interpretuota kaip klaida, mat informantai mėgina kurti atsakymus, kadangi to jų prašo interviu ėmėjas. Pavyzdžiui, klausiamas, kodėl vertina vienaip, o ne kitaip, informantas gali pateikti neišsamų, paviršutiniškų priežasčių.

Tyrėjos dalyvavimas, renkant veiklos tyrimo duomenis, viena vertus, yra teigiamas, nes tyrėjas nėra nešališkas, atitolęs stebėtojas; kita vertus, tai apribojimas, mat tiriamieji gali neatskleisti tyrėjai su asmenybe susijusių ir tyrimui įtakos turinčių veiksmų. Kitas kokybinio tyrimo apribojimas susijęs su taikytų mokymosi metodų vertinimais. Akivaizdu, kad mokymosi metodų vertinimams įtakos turi ne tik mokymosi metodų ir mokymosi stilių dermė, bet ir kiti veiksniai: mokymosi motyvacija, mokymosi aplinka, dėstytojo asmenybė ir kt. Jų poveikis taip pat neretai gali būti didesnis nei taikomo mokymosi metodo ar studentui būdingo mokymosi stiliaus. Šiuos apribojimus bandyta įveikti, interviu pradžioje išsamiai instruktuojant interviu davėjus.

⁹ Nacionalinis studijų kultūros kaitos projektas „Europos kreditų perkėlimo ir kaupimo sistemos (ECTS) nacionalinės koncepcijos parengimas: kreditų harmonizavimas ir mokymosi pasiekimais grindžiamų studijų programų metodikos kūrimas ir diegimas“; http://www.ects.cr.vu.lt/Projekto_rezultatai.

3. STUDENTŲ PROGRAMAVIMO GEBĖJIMŲ UGDYMO SI, ĮVERTINANT MOKYMO SI STILIŲ IR MOKYMO SI METODŲ DERMĖ, TYRIMO EMPIRINIS PAGRINDIMAS

3.1. ŽVALGOMOJO TYRIMO REZULTATAI

3.1.1. Programavimo gebėjimų ugdymo programos Lietuvos aukštosiose mokyklose

Programavimo mokymosi paradigmos ir programavimo kalbos pasirinkimas.

Programavimo mokymosi tyrėjai išskiria keletą programavimo mokymosi paradigmu, kuriose akcentuojama skirtinga programavimo mokymosi pradžia. Atlikta studiju programų ir įvadinio programavimo kursų aprašu analizė parodė (žr. 18 lentelę), kad programavimo studijos Lietuvos aukštosiose mokyklose dažniausiai pradeda mos nuo procedūrinio programavimo paradigmos (48,65 proc.). Kitas kiek retesnis pasirinkimas – pradeda nuo procedūrinio programavimo, o paskui įveda mos objektinio programavimo sąvokos. Objektinio programavimo paradigma nustatyta 8 ĮPK aprašuose, tačiau 5 iš jų reikalaujama būtino pasirengimo kurso studijoms, t. y. pradiniu programavimo žiniu, įgytu bendrojo ugdymo mokykloje. Bendrojo ugdymo mokyklose įgyjami būtent procedūrinio programavimo pradmenys. Taigi šiuo atveju galima teigti, kad studijos pradeda mos nuo objektinio programavimo pa radigmos, tačiau jau turima procedūrinio programavimo pradmenų. Užfiksuoti du atvejai, kai programavimo studijos pradeda mos nuo matematinės logikos ir loginio programavimo paradigmos (universitetinė studiju programa), ir vienas atvejis, kai programavimo studijos pradeda mos nuo algoritmu (universitetinė studiju programa). Kitų programavimo mokymosi paradigmu ĮPK aprašuose identifikuota nebuvo.

Pasirinkimas pradėti programavimo studijas nuo procedūrinio programavimo susiklostęs istoriškai, mat dauguma dėstytoju ėmėsi dėstyti programavimo studijas būtent taip ir, remdamiesi savo asmenine patirtimi, tokią pradžią laiko tinkama. Kita priežastis, lemianti šios paradigmos pasirinkimą, – jos paprastumas. Mokslinėje lite ratūroje plačiai diskutuojama apie programavimo studiju privalumus ir trūkumus, kai pradeda nuo procedūrinio, o nuo objektinio programavimo paradigmos. Viena iš labiausiai motyvuojančiu priežastiu – šios paradigmos populiarumas praktinėje veikloje. Kaip dar viena priežastis nurodoma, kad pradedantiesiems sunkumu kyla būtent dėl paradigmu kaitos. Manoma, kad būtų galima išvengti keblumu, kylančiu

turintiesiems procedūrinio programavimo patirties, jei programavimo studijos būtų pradedamos nuo objektinio programavimo paradigmos.

18 lentelė

Programavimo mokymosi paradigmos Lietuvos aukštosiose mokyklose

Programavimo mokymosi paradigma	Kolegijos		Universitetai		Iš viso	
	Kiekis	Proc.	Kiekis	Proc.	Kiekis	Proc.
Pradedama nuo procedūrinio programavimo	10	27,03	8	21,62	18	48,65
Pradedama nuo procedūrinio programavimo ir paskui įvedamos objektinio programavimo sąvokos	3	8,11	10	27,03	13	35,14
Pradedama nuo objektinio programavimo	2	5,41	1	2,70	3	8,11
Pradedama nuo loginio programavimo	0	0	2	5,41	2	5,41
Pradedama nuo algoritmų	0	0	1	2,70	1	2,70
Iš viso	15	40,54	22	59,46	37	100,00

Kitas aktualus įvadinio programavimo mokymosi tyrėjams klausimas – pirmosios programavimo kalbos pasirinkimas. Mokantis pagrindinių programavimo sąvokų nėra labai svarbu pasirinkti kalbą, tačiau jeigu tarp studijų programos tikslų numatoma ugdytis programavimo kompetenciją, turi būti mokomasi ir programavimo kalbos. Remiantis atlikta ĮPK aprašų analize galima teigti, kad įprastai šiems dalykams dėstyti naudojama C++ programavimo kalba (žr. 19 lentelę; 59,46 proc.).

19 lentelė

Programavimo kalbos pasirinkimas Lietuvos aukštosiose mokyklose

Programavimo kalba	Kolegijos		Universitetai		Iš viso	
	Kiekis	Proc.	Kiekis	Proc.	Kiekis	Proc.
C	1	2,70	3	8,11	4	10,81
C++	7	18,92	15	40,54	22	59,46
Java	1	2,70	0	0	1	2,70
Nėra konkrečios programavimo kalbos	0	0	2	5,41	2	5,41
Pascal	1	2,70	2	5,41	3	8,11
PHP	3	8,11	0	0	3	8,11
Visual Basic	1	2,70	0	0	1	2,70
Visual Basic for Application	1	2,70	0	0	1	2,70
Iš viso	15	40,54	22	59,46	37	100,00

Pirmosios programavimo kalbos pasirinkimas pagal studentų (2013 m.) skaičių

Pirmoji programavimo kalba IPK aprašuose	Kolegijose		Universitetuose		Iš viso	
	Studentų skaičius	Proc.	Studentų skaičius	Proc.	Studentų skaičius	Proc.
Pascal	41	4,2	163	13,5	204	9,3
VBA	56	5,8	0	0	56	2,6
C++	113	11,6	611	50,5	724	33,2
C	321	33,1	196	16,2	517	23,7
Java	58	6,0	0	0	58	2,7
PHP	178	18,3	0	0	178	8,2
Python	100	10,3	0	0	100	4,6
C++, VB	26	2,7	0	0	26	1,2
Nėra konkrečios PK	0	0,0	212	17,5	212	9,7
Nėra duomenų	78	8,0	29	2,4	107	4,9
Iš viso	971	100,0%	1211	100,0%	2182	100,0%

Įvertinkime pirmosios programavimo kalbos pasirinkimą studijuojančiųjų skaičiaus atžvilgiu. Remiantis studentų registro duomenimis, IK krypties studijas 2013 metais pasirinko 2 182 studentai. Apžvelgus stojančiųjų pasirinktas studijų programas nustatyta (žr. 20 lentelę), kad daugiausiai studentų programavimo studijas pradeda nuo C++ (33,2 proc.) ir C (23,7 proc.) programavimo kalbų. Be konkrečios programavimo kalbos pradeda 9,7 proc. studentų, o programavimo kalba *Pascal* – 9,3 proc. studentų. Pastaraisiais metais auga internetinio programavimo populiarumas, todėl net 8,2 proc. studentų programavimo studijas pradeda nuo *PHP* programavimo kalbos. Kitų programavimo kalbų vartojama labai nedaug.

Renkant duomenis apie programavimo kalbos pasirinkimą aukštosiose mokyklose nustatyta, kad kolegijose IPK studijoms dažniau pasirenkama C (33,1 proc.) programavimo kalba, o universitetuose – C++ (50,5 proc.). Pastebima, kad kolegijose didesnė pirmosios programavimo kalbos pasirinkimo įvairovė, atsiradusi su išaugusiu naujų šiuolaikiškų studijų programų pasirinkimu. Universitetinėse studijų programose 17,5 proc. studentų studijas pradeda be konkrečios programavimo kalbos, tačiau kolegines studijų programose tokių atvejų nenumatyta. Renkant pirmąją programavimo kalbą kolegijose daugiau reikšmės turi programavimo kalbos populiarumas ir pritaikomumas profesinėje veikloje. Disertantės nuomone, per trejus ar ketverius studijų metus nereikėtų siekti išmokyti kuo daugiau skirtingų programavimo kalbų, verčiau gilintis į vieną ar dvi pasirinktas praktinio programavimo kryptis.

Remiantis atlikta ĮPK aprašų analize ir nustatčius studijų programose studijuojančių studentų skaičių, galima teigti, kad daugiausia studentų įvadinio programavimo mokymą pradeda nuo C ir C++ programavimo kalbų. Šiais atvejais objektinio programavimo paradigma įtraukiama kurso studijų pabaigoje arba apsiribojama tik procedūrinio programavimo paradigma.

ĮPK tikslai, numatyti įvairiose IK kryptų studijų programose, skiriasi – kurso tikslai ir siekiami rezultatai derinami su visos studijų programos siekiniais. Analiizuojant šiuos tikslus, pastebėta, kad dalis iš jų akcentuoja programavimo paradigmą: „*Supažindinti studentus su pagrindiniais struktūrinio programavimo principais, apžvelgti objektinio programavimo koncepcijas.*“ Kiti – siekį išmokyti studijuojamą programavimo kalbą: „*Dalyko tikslas – suteikti programavimo C++ kalba žinių bei praktinių įgūdžių, būtinų dirbti su įvairiais duomenimis, kurti taikomąsias programas.*“ Tik 6 ĮPK aprašuose minimas gebėjimo analizuoti užduotį ugdymasis: „*Studentai gebės analitiškai mąstyti ir konkretinti uždavinius*“; „*Nustatys pradinis duomenis, pagrindinius bei pagalbinis kintamuosius bei numatomus rezultatus*“ ir pan. Kurso studijų tikslai buvo įvertinti (žr. 21 lentelę) remiantis apibrėžta pradedančiųjų programavimo gebėjimų ugdymosi struktūra ir Bloomo ugdymo tikslų taksonomija. Nustatyta, kad dažniausiai tiksluose minima pasirinkta programavimo paradigma, programavimo ir / arba algoritmavimo gebėjimų ugdymas, programavimo kalbos sintaksės žinios. Kai kuriais atvejais akcentuojamas tik gebėjimas taikyti studijų metu įgytas žinias, o dar kitais atvejais minimas ir kūrybiškumo ugdymas.

Keliant įvadinio programavimo kurso tikslus nereikėtų apsiriboti tik programavimo konceptų supratimu ir programavimo kalbos sintaksės žiniomis, mat labai svarbu yra ugdytis ir uždavinių sprendimo bei programavimo gebėjimus. Vis dėlto ĮPK aprašuose nurodyti tikslai dažniau akcentuoja paradigmą realizuojančios programavimo kalbos studijas ir atskirai neišskiriamas užduočių analizės ir problemų sprendimo gebėjimų ugdymasis.

ĮPK tikslų analizė

Kategorija	Subkategorija	Pasikartojimo tiksluose dažnis	
		Dažnis	Proc.
Programavimo procesas	Programavimo konceptai: algoritmai, jų kūrimas ir taikymas; duomenų tipai; duomenų struktūros	23	79,31
	Uždavinių analizės ir sprendimo gebėjimai	11	37,93
	Programavimo kalbos sintaksė ir sintaksės žinių taikymas	14	48,28
	Derinimas ir testavimas	6	20,69
Programavimo paradigma	Struktūrinis programavimas	7	24,14
	Objektinis programavimas	8	27,59
Bloomo taksonomijos lygmenys	Žinios	29	100
	Supratimas	25	86,21
	Taikymas	25	86,21
	Analizė	17	58,62
	Vertinimas	17	58,62
	Kūryba	16	55,17
ĮPK aprašų skaičius		29	

ĮPK tematika. Siekiant įvertinti ĮPK aprašuose išskiriamą tematiką, peržiūrėtas įvadinio programavimo kurso turinys, kurį kaip pavyzdį rekomenduoja tarptautinės asociacijos ACM ir IEEE, rengiančios rekomendacijas (anglų k. *Computing Curriculum*) IK krypties studijoms. Pavyzdinį įvadiniam programavimui mokytis skirtą kursą „Įvadas į programavimą“ (anglų k. *Introduction to Programming*) parengė D. Reedas (Kreitono universitetas). Studijų programoje tai pirmasis programavimo kursas, jį sudaro temos iš keturių CSC2013 apibrėžtų žinių sričių: 1) Programinės įrangos kūrimo pagrindai (SDF), 2) Programavimo kalbos (PL), 3) Algoritmai ir jų sudėtingumas (AL), 4) Socialinės problemos ir profesinė praktika (SP). Kursas dėstomas vartojant *Python* programavimo kalbą. D. Reedas programavimo studijas dėsto nuo procedūrinio programavimo (kintamieji, išraiškos, funkcijos, valdymo struktūros ir kt.), paskui pereina prie objektinio programavimo įvado (objektai, klasės, metodai, simbolių eilutės, duomenų failai ir kt.). Šių temų dėstymo eiliškumą D. Reedas pasirinko remdamasis savo ankstesne patirtimi: bandyta dėstyti šį dalyką *Java* programavimo kalba ir pradėdant nuo objektinio programavimo paradigmos, tačiau, D. Reedo nuomone, tai pernelyg sudėtinga pradėdantiesiems.

Tolimesnei lyginamajai analizei atrinkti ĮPK aprašai (9 universitetinių studijų ĮPK aprašai; 13 koleginių studijų ĮPK aprašų), kuriuose taip pat numatoma procedūrinio programavimo mokymosi paradigma. O likusieji ĮPK aprašai tematikos kontekste

nebuvo analizuojami ir lyginami su minėtu pavyzdiniu aprašu, nes juose naudojamos kitos programavimo mokymosi paradigmos.

Palyginus Lietuvos aukštosiose mokyklose dėstomų ĮPK tematiką su CSC2013 pateiktu pavyzdiniu kursu, nustatyta (žr. 22 lentelę), kad daugiau kaip pusę (tačiau ne daugiau kaip 72 procentus) pavyzdinio kurso temų atitinka tik penki analizuoti ĮPK aprašai. Kituose aprašuose sutampančių temų mažiau. Apžvelgiant kursų tematiką pastebėta, kad daugiausia dėmesio skiriama paradigmą realizuojančios programavimo kalbos studijoms, neskiriama laiko algoritams, uždavinių sprendimo strategijoms, programavimo metodams, saugiam duomenų tipų naudojimui, tinkamų duomenų struktūrų parinkimui, programavimo kultūrai ir kt.

22 lentelė

Kurso „Įvadas į programavimą“ (Kreitonos universitetas) ir analizuotų ĮPK aprašų tematikos palyginimas

CSC 2013 Žinių sritis, „Įvado į programavimą“ (Kreitonos universitetas) temos	Temų pasikartojimo dažnis tirtuose ĮPK aprašuose	
	Kolegijos (13)	Universitetai (9)
SDF: Algoritmai ir jų sudarymas		
1. Pagrindinės sąvokos ir algoritmų savybės	6	6
2. Algoritmų vaidmuo sprendžiant uždavinius	6	2
3. Uždavinių sprendimo strategijos (pasikartojimas, „skaldyk ir valdyk“)	0	0
4. Algoritmų įgyvendinimas	2	3
5. Algoritmų sudarymo principai (abstrakcija, dekompozicija)	1	0
SDF: Pagrindinės programavimo sąvokos		
1. Sintaksė ir semantika	13	8
2. Kintamieji ir primitivai	13	8
3. Išraiškos ir priskyrimas	13	8
4. Paprasta įvestis ir išvestis	13	8
5. Sąlygos ir iteracijos	13	8
6. Funkcijos ir parametrai	13	8
SDF: Pagrindinės duomenų struktūros		
1. Masyvai	10	9
2. Įrašai	10	5
3. Simbolių eilutės	9	5
4. Tinkamų duomenų struktūrų parinkimo strategijos	2	3
SDF: Programavimo metodai		
1. Programų teisingumas (specifikacija, apsaugos programavimas, testavimo pagrindai)	3	1
2. Šiuolaikinės aplinkos	3	0
3. Derinimo strategijos	4	1
4. Dokumentavimas ir programavimo stilius	2	1

CSC 2013 Žinių sritis, „Įvado į programavimą“ (Kreitonos universitetas) temos	Temų pasikartojimo dažnis tirtuose ĮPK aprašuose	
	Kolegijos (13)	Universitetai (9)
PL: Objektinis programavimas		
1. Objektinis projektavimas	0	0
2. Klasės ir objektai	5	0
3. Laukai ir metodai	5	0
PL: Pagrindinių tipų sistemos		
1. Paprastieji duomenų tipai	13	0
2. Tipų saugumas ir klaidos	8	0
PL: Kalbos transliatorius		
1. Interpretavimas	0	0
2. Vertimo procesas (anglų k. <i>translation pipeline</i>)	0	0
AL: Duomenų struktūrų ir algoritmų pagrindai		
1. Paprasti skaitiniai algoritmai	10	5
2. Nuosekli paieška	8	5
3. Paprastų eilučių apdorojimas	10	4

ĮPK studijų rezultatai – tai „teiginiai, kurie nurodo, ką studentas žino, supranta ir yra pajėgus padaryti pasibaigus mokymosi arba studijų procesui, ir yra apibrėžiami žiniomis, mokėjimais ir kompetentingumu“ (European Qualification Framework, 2008). Tolimesnei kurso studijų rezultatų analizei pasirinkti ĮPK aprašai (9 universitetinių studijų ĮPK aprašai; 13 koleginių studijų ĮPK aprašų), numatantys procedūrinio programavimo mokymosi paradigmą. Analizės metu apibendrinti kurso studijų rezultatai – aprašomos kursą išklausiūsi studento žinios ir gebėjimai (žr. 23 lentelę). Šiuose ĮPK aprašuose vyrauja studijų rezultatai, įvardijami kaip žinių supratimas ir jų taikymas, rečiau aprašomas analitinių ir kūrybinių gebėjimų ugdymasis.

23 lentelė

ĮPK studijų rezultatai

Kurso studijų rezultatai	Pasikartojimų dažnis ĮPK aprašuose		
	Nurodytas	Gali būti įvardytas	Nenurodytas
Žinos programavimo kalbų istoriją ir jų vystymosi tendencijas	2	11	2
Žinos programų kūrimo etapus	4	5	6
Žinos pagrindines programavimo sąvokas	1	14	0
Žinos programavimo kalbos leksiką ir sintaksę	6	9	0
Žinos bazinius duomenų tipus ir gebės jais naudotis	8	7	0
Supras struktūrinius duomenų tipus	2	10	3

Kurso studijų rezultatai	Pasikartojimų dažnis ĮPK aprašuose		
	Nurodytas	Gali būti įvardytas	Nenurodytas
Supras masyvus ir gebės jais naudotis	5	8	2
Gebės kurti algoritmus, apdorojančius eilutės tipo duomenis	1	7	7
Gebės naudoti programose struktūras (įrašus)	1	7	7
Supras programos kintamųjų prieinamumo lygius (tinkamai naudos globalius, lokalius, statinius kintamuosius)	5	7	3
Mokės kurti naujas duomenų struktūras	3	2	10
Gebės naudotis rodyklėmis	1	5	9
Gebės taikyti dinامينius kintamuosius	2	4	9
Gebės derinti duomenų tipus programinėse konstrukcijose	2	8	5
Gebės dirbti su duomenų failais	6	8	1
Gebės taikyti ir kurti tiesinius ir šakotuosius algoritmus	10	4	1
Gebės skaidyti užduotį į dalis, aprašyti funkcijas ir jomis naudotis	8	5	2
Supras, taikys ir gebės kurti ciklinius algoritmus	9	5	1
Gebės analizuoti užduotį ir parengti jos realizavimo algoritmą	11	0	4
Gebės užrašyti algoritmą programavimo kalba	9	5	1
Gebės skaityti ir paaiškinti algoritmą	3	1	11
Gebės testuoti ir derinti sukurtas programas	10	2	3
Gebės dokumentuoti programas	2	2	11
Suvoks algoritmų paskirtį ir savybes	1	6	8
Žinos pagrindines objekcinio programavimo sąvokas	6	4	5
Supras objekcinio programavimo savybes: moduliškumą, polimorfizmą ir paveldimumą	4	2	9
Gebės taikyti ir kurti objektus, aprašyti klasę ir ja naudotis	5	4	6
Gebės įdiegti programavimo aplinką	2	0	13
Gebės sukurti savo programos diegimo paketą	2	0	13
Gebės skaityti profesinę literatūrą	4	2	9
Gebės savarankiškai gilinti ir plėsti gautas žinias	4	4	7
Gebės projektuoti, kurti ir tobulinti taikomųjų programų vartotojo sąsają	2	2	11
Gebės taikyti rekursinius algoritmus	0	4	11
Žinos ir laikysis programavimo kultūros taisyklių	0	1	14
Žinos populiariausius algoritmų vaizdavimo būdus	2	1	12

Kaip minėta anksčiau, tarp studijų kurso tikslų bene visai neminimas uždavinių analizės, sudėtingesnių užduoties sprendimo būdų paieškos bei realizavimo gebėjimų ugdymasis, tačiau net vienuolikoje dalykų aprašų tarp kurso studijų rezultatų įvardytas gebėjimas analizuoti užduotį ir parengti jos sprendimo algoritmą. Taigi galima teigti, kad IK krypčių studijų programose šis gebėjimas pripažįstamas kaip labai svarbus, tiesiog nėra nurodomas kaip ĮPK tikslas. Studijų rezultatai, susiję su studijuojamos

programavimo kalbos sintakse ir semantika, įvardyti beveik visuose analizuotuose ĮPK aprašuose. Rečiau sutinkami studijų rezultatai, susiję su šiais gebėjimais:

- kurti naujas duomenų struktūras (3 aprašuose įvardytas, 2 aprašuose įtraukta į temų sąrašą),
- naudoti rodykles (1 apraše įvardytas, 5 aprašuose įtraukta į temų sąrašą),
- taikyti dinامينius kintamuosius (2 aprašuose įvardytas, 4 aprašuose įtraukta į temų sąrašą).

Šėšiuose iš analizuotų ĮPK aprašų skiriama paskaitų objekcinio programavimo temoms, tačiau studijų rezultatų, susijusių su šia tematika, suformuluota nėra arba yra po vieną du. Tarp kurso studijų rezultatų minimas bendrasis gebėjimas savarankiškai gilinti ir plėsti gautas žinias yra ypač aktualus mokymosi visą gyvenimą aspektu. Viena iš šiuolaikinio programuotojo atliekamų užduočių – rengti įvairią dokumentaciją, tad svarbu ugdytis gebėjimą dokumentuoti programas; vis dėlto jis nurodytas tik dviejuose ĮPK aprašuose. Jį taip pat būtų galima priskirti ir prie dar dviejų kursų, kurių studijų metodų sąrašė numatomas rašto darbas – parengti sukurtos programos ataskaitą.

Taigi ne visuose aprašuose išsamiai atskleisti studijų rezultatai, kai kuriuose jie labai apibendrinti. Pastebėta, kad didelės dalies ĮPK aprašų tematika rodo, kad tam tikri studijų rezultatai yra, tačiau jie turėtų būti konkrečiau įvardijami ir / ar detalizuojami.

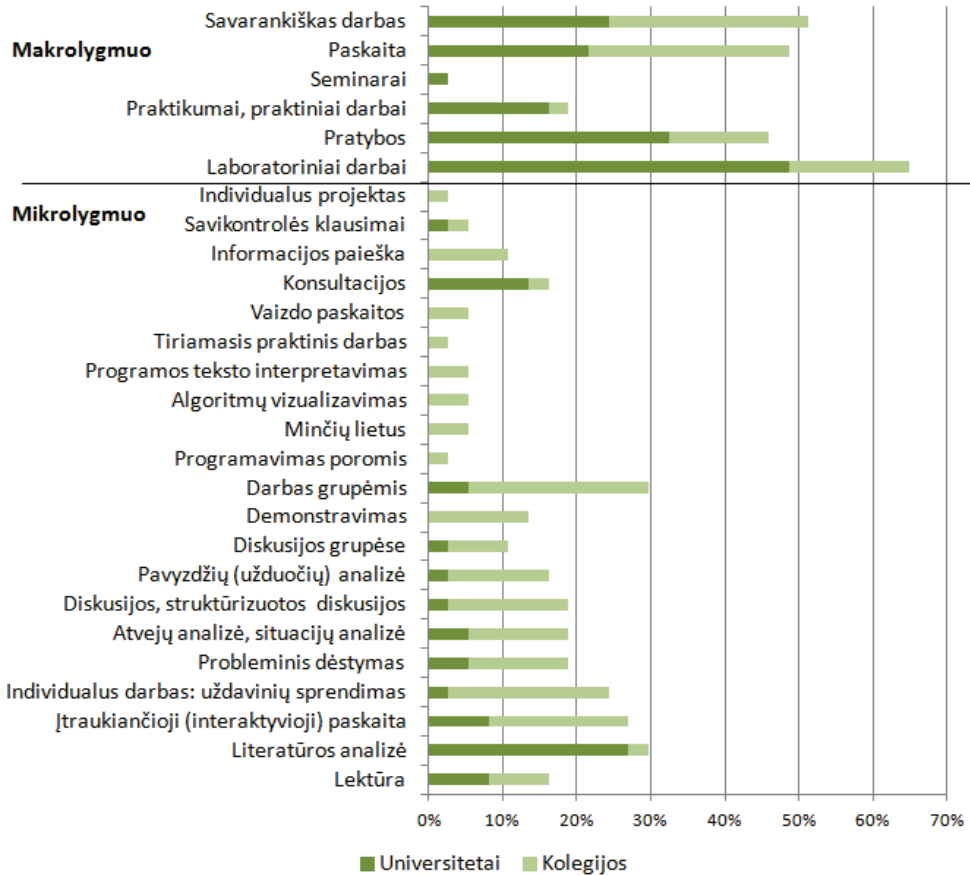
Studijų ir vertinimo metodai. Kadangi studijų procese taikomi studijų ir vertinimo metodai nepriklauso nuo programavimo mokymosi paradigmos ar pasirinktos programavimo kalbos, šiuo atžvilgiu buvo analizuojami visi pagal reikalavimus parengti ĮPK aprašai. Atmesti tik 6 aprašai, kuriuose studijų ir vertinimo metodai nenurodyti.

Studijų metodai – tai įvairios mokymosi veiklos, kuriose dalyvauja studentai ir dėstytojai. Ir Lietuvoje, ir užsienyje įvadinio programavimo studijoms taikomi įvairūs studijų metodai, kadangi dėstomos įvairios programavimo kalbos, naudojamos skirtingos programavimo mokymosi paradigmos ir strategijos (Van-Roy, Haridi, 2004; Vitkutė-Adžgauskienė, Vidžiunas, 2012). Akivaizdu, kad ĮPK apraše nurodomi makrolygmens studijų metodai visiškai neatskleidžia studijų metu atliekamos dėstytojo ir studentų veiklos. Tad šis studijų metodų skirstymas pasiteisina tik studijų organizavimo aspektu (Teresevičienė ir kt., 2011).

ĮPK aprašuose studentų darbo valandos paskirstomos kontaktiniam darbui auditorijoje (paskaitoms, laboratoriniams darbams, pratyboms ir pan.) ir savarankiškam darbui. Kolegijose vienam kreditui skiriama vidutiniškai 14,3 kontaktinio darbo valandų ir 12,7 savarankiško darbo valandų; universitetuose – 11,7 kontaktinio dar-

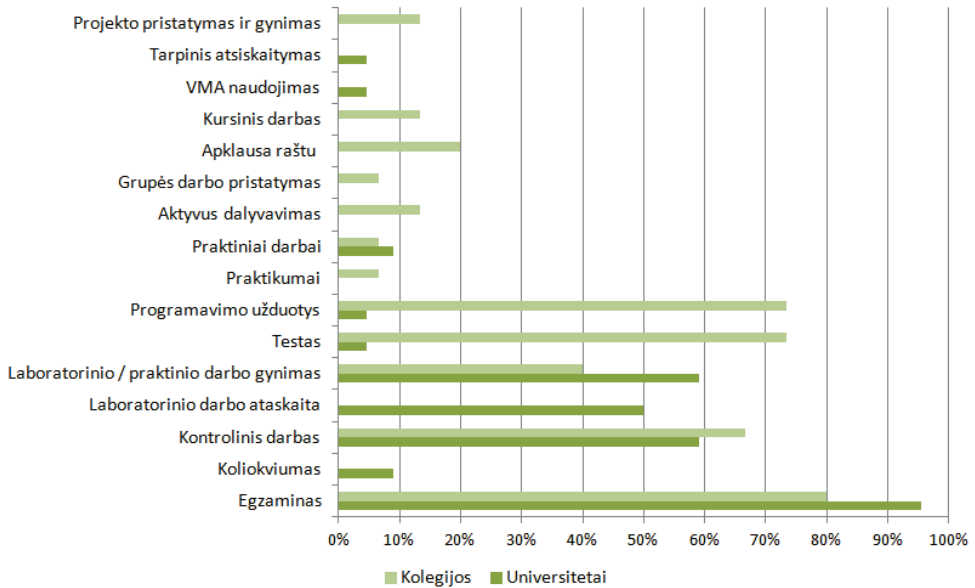
bo valandų ir 15,7 savarankiško darbo valandų. Lyginant paskaitoms ir pratyboms (ar laboratoriniams darbams) skiriamą laiką nustatyta, kad kolegijose praktiniams gebėjimams ugdytis skiriama kiek daugiau laiko (kolegijose – vidutiniškai 8 valandas vienam kreditui; universitetuose – 6,5 valandų). Mokant programavimo ypač svarbus dėstytojo skiriamas laikas individualioms konsultacijoms, tačiau ĮPK aprašuose konsultacijų laikas paprastai nenurodomas.

Vertinant mikrolygmens studijų metodų pasikartojimą, pastebima, kad kolegijų ĮPK aprašuose nurodomi įvairesni studijų metodai. Be to, šiuose aprašuose didesnis ir šiuolaikiškų, ir aktyvaus darbo studijų metodų (darbas grupėse, diskusijos, minčių lietus, probleminis dėstymas ir kt.) dažnis (žr. 12 paveikslą).



12 pav. Studijų metodai ĮPK aprašuose

Analizuojant taikomus vertinimo metodus nustatyta, kad pradedančiųjų programavimo gebėjimai paprastai vertinami per egzaminą (žr. 13 paveikslą). Jo metu tikrinamos ir įgytos teorinės žinios, ir gebėjimas spręsti programavimo užduotis. Kiti dažniau taikomi vertinimo metodai – laboratoriniai darbai, laboratorinių darbų gynimas ir kontroliniai darbai.



13 pav. Vertinimo metodai ĮPK aprašuose

Apibendrinant atliktą dokumentų turinio analizę galima teigti, kad:

- ĮPK aprašuose deklaruojama, jog Lietuvos aukštosiose mokyklose (kiek daugiau kaip du trečdaliai IK kryptių studijų programų) programavimo studijos pradedamos nuo procedūrinio programavimo paradigmos, vartojamos C arba C++ programavimo kalbos. Objektinio programavimo paradigma dažniausiai įvedama ĮPK pabaigoje arba kituose programavimo kursuose prieš tai įgijus procedūrinio programavimo žinių;
- Nurodant ĮPK tikslus dažniau įvardijama pasirinkta programavimo paradigma, programavimo ir / arba algoritmavimo gebėjimų ugdymasis, programavimo kalbos sintaksės žinių suteikimas, rečiau – kiti programavimo proceso etapuose būtini gebėjimai: užduočių analizė, programavimo aplinkos naudojimas, derinimas ir testavimas. Analizuojant tikslus pagal Bloomo

taksonomiją pastebėta, kad didžioji dalis ĮPK apima žinojimo, supratimo ir taikymo lygmenis. Kiek daugiau kaip pusėje ĮPK aprašų tarp tikslų numatyta ir analizės, vertinimo bei kūrybinio lygmens siekis;

- Tik penki analizuoti Lietuvos aukštųjų mokyklų ĮPK aprašai atitinka daugiau kaip pusę, tačiau ne daugiau kaip 72 procentus Kreitono universiteto kurso (pateikiamo kaip pavyzdinio CSC2013 rekomendacijose) temų. Kituose aprašuose sutampančių temų mažiau. Beveik visuose Lietuvos aukštųjų mokyklų ĮPK aprašuose skiriama dėmesio paradigmą realizuojančios programavimo kalbos studijoms, rečiau – algoritmams, uždavinių sprendimo strategijoms, programavimo metodams, saugiam duomenų tipų naudojimui, tinkamų duomenų struktūrų parinkimui, programavimo kultūrai ir kt. temoms, aktualioms ugdantis pradedančiųjų programavimo gebėjimus;
- Mikrolygmens studijų metodai dažniau nurodomi kolegijų ĮPK aprašuose nei universitetų. Būtent todėl šiuolaikiškų ir aktyvaus darbo studijų metodų (darbas grupėse, diskusijos, minčių lietus, probleminis dėstymas ir kt.) dažnis didesnis kolegijų aprašuose. Pradedančiųjų programavimo gebėjimai vertinami taikant vertinimo metodus: egzaminą, kontrolinius darbus, laboratorinius darbus ir jų gynimą.

3.1.2. Kiekybinis tyrimas „Studentų požiūris į įvadinio programavimo mokymąsi“

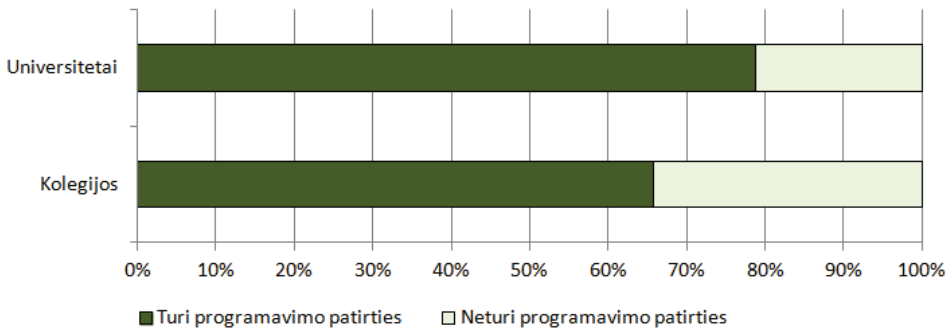
Anketinio tyrimo respondentų demografinės charakteristikos. Iš tyrimo dalyvavusių 377 dalyvių dauguma (77,5 proc.) buvo vyriškos lyties asmenys. Remiantis Lietuvos statistikos departamento duomenimis¹⁰, moteriškos lyties atstovių, įstojusių į technologijų ir fizinių mokslų sritis, skaičius sudaro apie trečdalį visų studijuojančiųjų (fiziniai mokslai: universitetuose 35,5 proc., kolegijose 9 proc.; technologiniai mokslai: universitetuose 19,5 proc., kolegijose 20,4 proc.). Atitinkamai ir IK kryptių studijas vyriškos lyties asmenys renkasi dažniau.

Daugiausia tyrimo dalyvavo 18–20 metų amžiaus asmenų (66,3 proc.). Šiek tiek mažiau negu trečdalis (27,6 proc.) tyrimo dalyvavusiųjų buvo 21–25 metų, o mažiausią respondentų dalį sudarė asmenys nuo 26 iki 38 metų.

¹⁰ Lietuvos statistikos departamentas (<http://www.stat.gov.lt>).

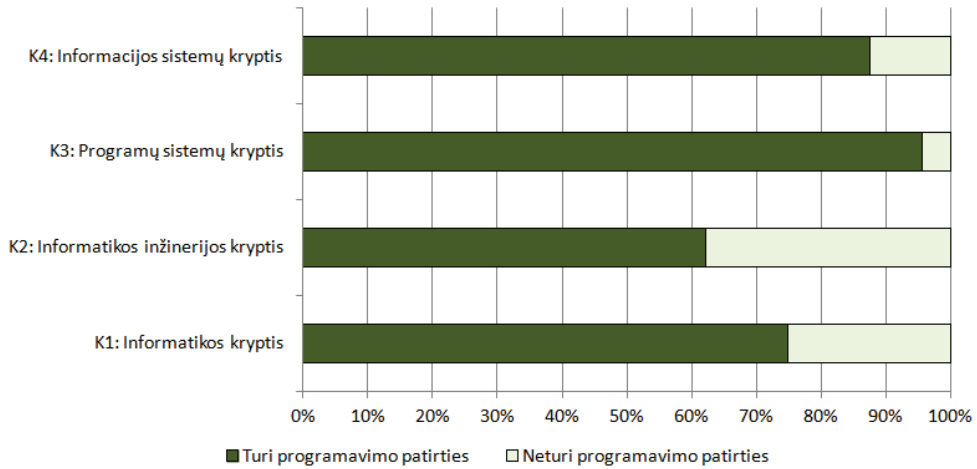
Geografinio pasiskirstymo požiūriu daugiausia apklaustųjų studijavo didžiuosiuose Lietuvos miestuose: Vilniuje (45,9 proc.) ir Kaune (20,7 proc.). Taip pat nemaža dalis asmenų apklausta iš Alytaus (19,4 proc.) ir Šiaulių (8,8 proc.). Nedidelę dalį (5,3 proc.) sudarė respondentai iš Klaipėdos, Panevėžio ir Utenos. Tyrimo dalyvių pasiskirstymas studijų vietos atžvilgiu gana įvairus.

Studentų pasirengimas programavimo studijoms. Anketinio tyrimo metu studentų buvo klausiama: „Ar anksčiau (iki studijų) teko susidurti su programavimu?“ Be jokios išankstinės programavimo patirties į įvadinio programavimo studijas tiriamuoju laikotarpiu susirinko kiek daugiau kaip ketvirtadalis (27,9 proc.) studentų, tarp jų 62,9 proc. kolegijų ir 37,1 proc. universitetų studentai. Nustatyta, kad kolegijose be išankstinės programavimo patirties studijas pradeda 13 proc. daugiau studentų negu universitetuose (žr. 14 paveikslą). Vadinasi, IK kryptčių studijas be pradinio pasirengimo dažniau pasirenka kolegijų studentai. Skirtumas statistiškai reikšmingas ($\chi^2 = 7,92$, $df = 1$, $p < 0,005$).



14 pav. Respondentų programavimo patirtis iki studijų aukštosiose mokyklose

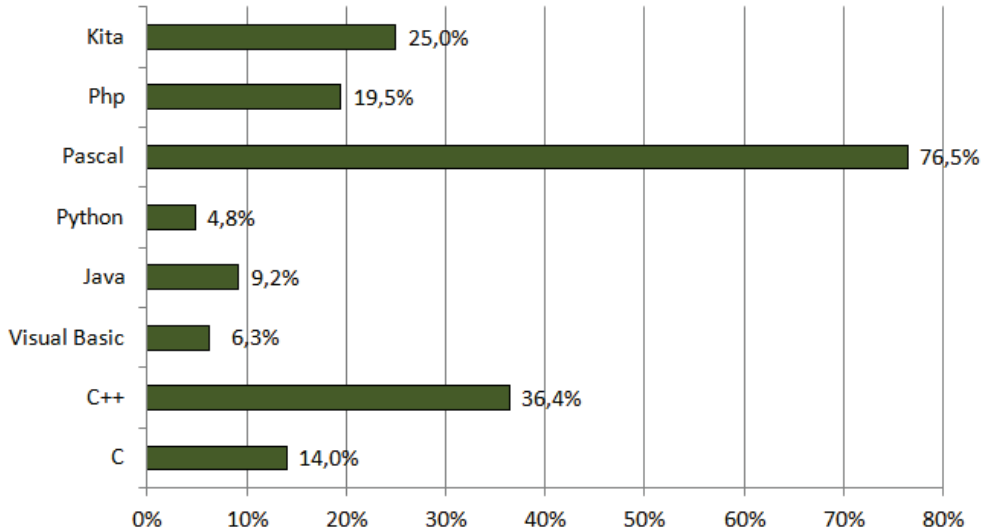
Analizuojant respondentų turimą programavimo patirtį pagal studijų kryptis (žr. 15 paveikslą), pastebėta, kad be pasirengimo dažniausiai pasirenkamos informatikos inžinerijos (37,8 proc.) ir informatikos kryptčių (25,2 proc.) studijos. Turintieji programavimo patirties dažniausiai renkasi programų sistemų kryptties (95,5 proc.) studijų programas. Lyginant respondentus pagal turimą programavimo patirtį ir studijų programų kryptis, nustatytas reikšmingas skirtumas ($\chi^2 = 33,38$, $df = 4$, $p < 0,001$).



15 pav. Respondentų programavimo patirtis iki studijų pagal studijų kryptis

Pasidomėjus, kur respondantai įgijo programavimo patirties, nustatyta, kad dauguma (88,6 proc.) programavimo pradėjo mokytis mokykloje. Maždaug kas trečias respondentas (36,4 proc.) prieš pradėdamas studijuoti savarankiškai mokėsi programuoti. Mažiau kaip dešimtadalis programuoti mokėsi įvairiose neformalaus mokymosi įstaigose, pvz., neakivaizdinėje programuotojų mokykloje (3,3 proc.). Respondantai minėjo ir kitus neformalaus mokymosi būdus: „Atlikau praktiką Darnijos įmonėje“; „Dalyvauju laisvos programinės įrangos kūrime, judėjime“; „Redagavau žaidimo kodą“; „Bandžiau programuoti žaidimo serverį“ ir pan. Analizuojant turimą patirtį pagal aukštąsias mokyklas ir studijų programų kryptis, nustatyta, kad patirties įgijimo būdas renkantis studijas įtakos neturėjo.

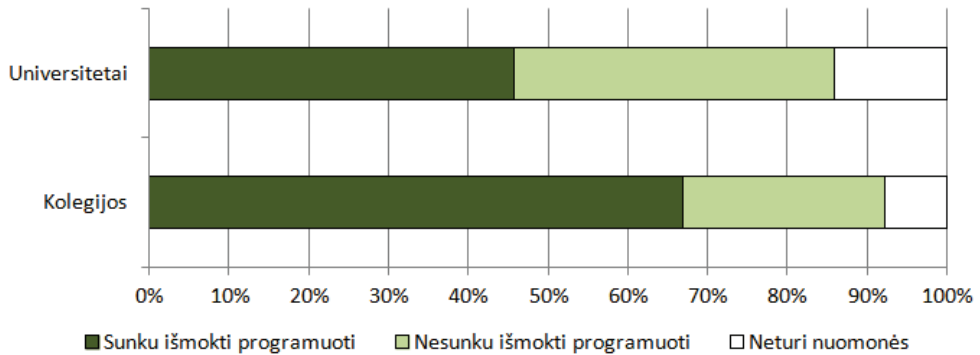
Domėtasi, su kokiomis programavimo kalbomis tiriamieji yra susipažinę ir moka programuoti (žr. 16 paveikslą). Reikia pastebėti, kad *Pascal* (76,5 proc.) ir *C++* (36,4 proc.) programavimo kalbų pasirinkimą lemia jų vartojimas bendrojo ugdymo mokyklose mokantis programavimo pradmenų, organizuojant šalies ir tarptautines informatikos olimpiadas ir laikant informacinių technologijų brandos egzaminą. Anketinės apklausos rezultatai rodo, kad respondentų turima programavimo patirtis labai įvairi, nurodyta ir kitų programavimo kalbų (*C*, *C#*, *PHP*, *Java*, *JavaScript*, *Visual Basic*, *Python* ir kt.).



16 pav. Respondentų turima programavimo patirtis pagal programavimo kalbas iki studijų

Studentų požiūris į programavimo mokymąsi ir išklauso tyto dalyko vertinimas.

Kadangi studentų nuostatos – tai įsitikinimai ir jausmai, kurie skatina tam tikru būdu reaguoti į studijų dalyką ir yra susiję su mokymosi motyvacija, anketose buvo prašoma išsakyti nuomonę apie programavimą. Dviejų trečdalių respondentų nuostata buvo teigiama: programuoti patinka daugumai (63,1 proc.) studijuojančiųjų. Tačiau nemaža dalis respondentų (21,1 proc.) teigė nemėgstantys programuoti. Reikšmingo skirtumo tarp universitetų ir kolegijų studentų atsakymų nenustatyta. Vertinant respondentų atsakymus pagal studijų kryptis nustatytas statistiškai reikšmingas skirtumas ($\chi^2 = 37,05$, $df = 4$, $p < 0,0001$), rodantis, kad programų sistemų studijas dažniausiai (89,6 proc.) renka teigiamą nuostatą programavimo atžvilgiu turintys studentai. Gana dažnai (31,6 proc.) informatikos inžinerijos ir rečiau (14,5 proc.) informatikos krypties studijas pasirenka studentai, kurių nuostata programavimo atžvilgiu yra neigiama.



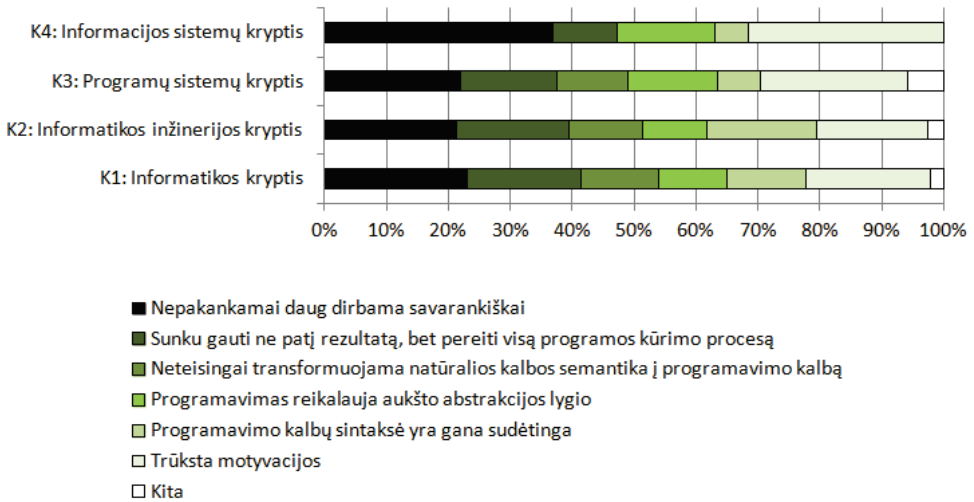
17 pav. Studentų nuomonė apie programavimo mokymosi sudėtingumą aukštosiose mokyklose

Siekiant nustatyti, ar respondentams sunku mokytis programuoti, į anketoje pateiktą klausimą „Ar, tavo manymu, sunku išmokyti programuoti?“ beveik trečdalis apklausoje dalyvavusių respondentų (32,6 proc.) atsakė, kad išmokyti programuoti yra lengva, kiek daugiau kaip pusė (56,5 proc.) – kad sunku. Lyginant universitetų ir kolegijų studentų atsakymus (žr. 17 paveikslą), nustatytas reikšmingas skirtumas ($\chi^2 = 17,34$, $df = 2$, $p < 0,001$). Du trečdaliai kolegijų studentų (66,8 proc.) programavimo mokymąsi laiko sudėtingu, o taip manančių universitetų studentų yra 21 procentu mažiau. Programavimo mokymosi sudėtingumo vertinimai pagal studijų kryptis labai neišsiskyrė, statistiškai reikšmingo skirtumo nenustatyta.

Pasidomėjus, kokios priežastys trukdė mokytis programuoti, respondentai buvo gana savikritiški (žr. 18 paveikslą). Daugiau kaip pusė nurodė savarankiško darbo (59,4 proc.) ir motyvacijos (53,6 proc.) trūkumą. Įvardytos ir tokios priežastys: programavimo kalbų sintaksės sudėtingumas (37,7 proc.), sunkumas ne parašyti programą, o praeiti visą jos kūrimo procesą (47,5 proc.).



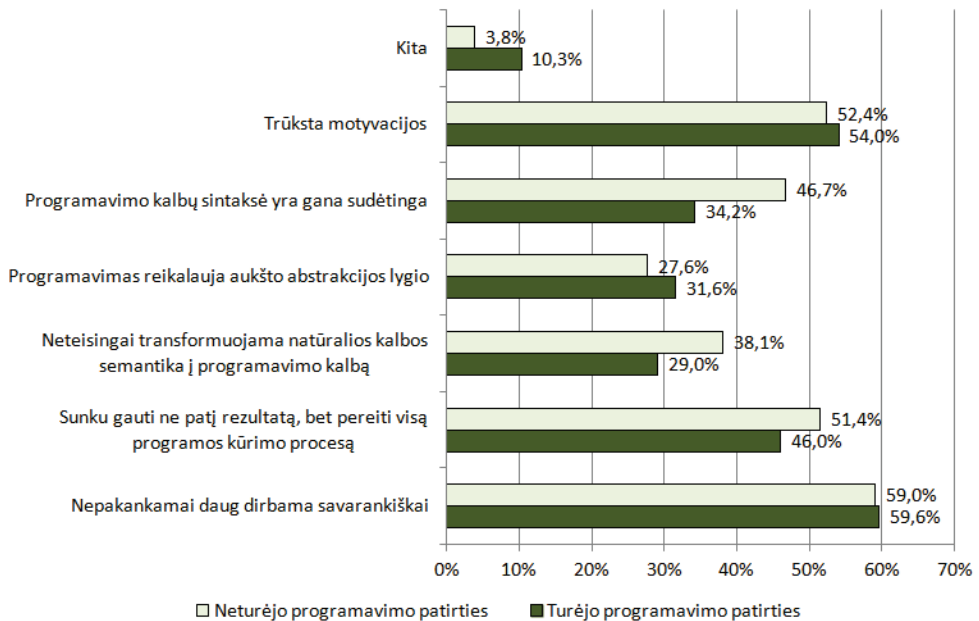
18 pav. Respondentų nurodomi programavimo mokymosi sunkumai



19 pav. Respondentų nurodomi programavimo mokymosi sunkumai pagal studijų kryptis

Analizuojant respondentų nurodomus programavimo mokymosi sunkumus pagal aukštąsias mokyklas ir studijų kryptis (žr. 19 paveikslą) reikšmingo nuomonių skirtumo nenustatyta.

Taip pat domėtasi, kokius programavimo mokymosi sunkumus nurodo neturintieji programavimo patirties, lyginant juos su respondentais, kurie studijas pradėjo turėdami patirties (žr. 20 paveikslą). Respondentų nuomonės šiek tiek išsiskyrė, kai jie vertino programavimo kalbos sintaksės ir natūralios kalbos semantikos transformavimo į programavimo kalbą sudėtingumą. Turintieji ilgesnę programavimo patirtį šiuos sunkumus nurodė apie 9–12 proc. punktų rečiau nei neturintieji ankstesnės patirties. Tačiau statistiškai reikšmingas ryšys tarp šių kintamųjų nenustatytas.

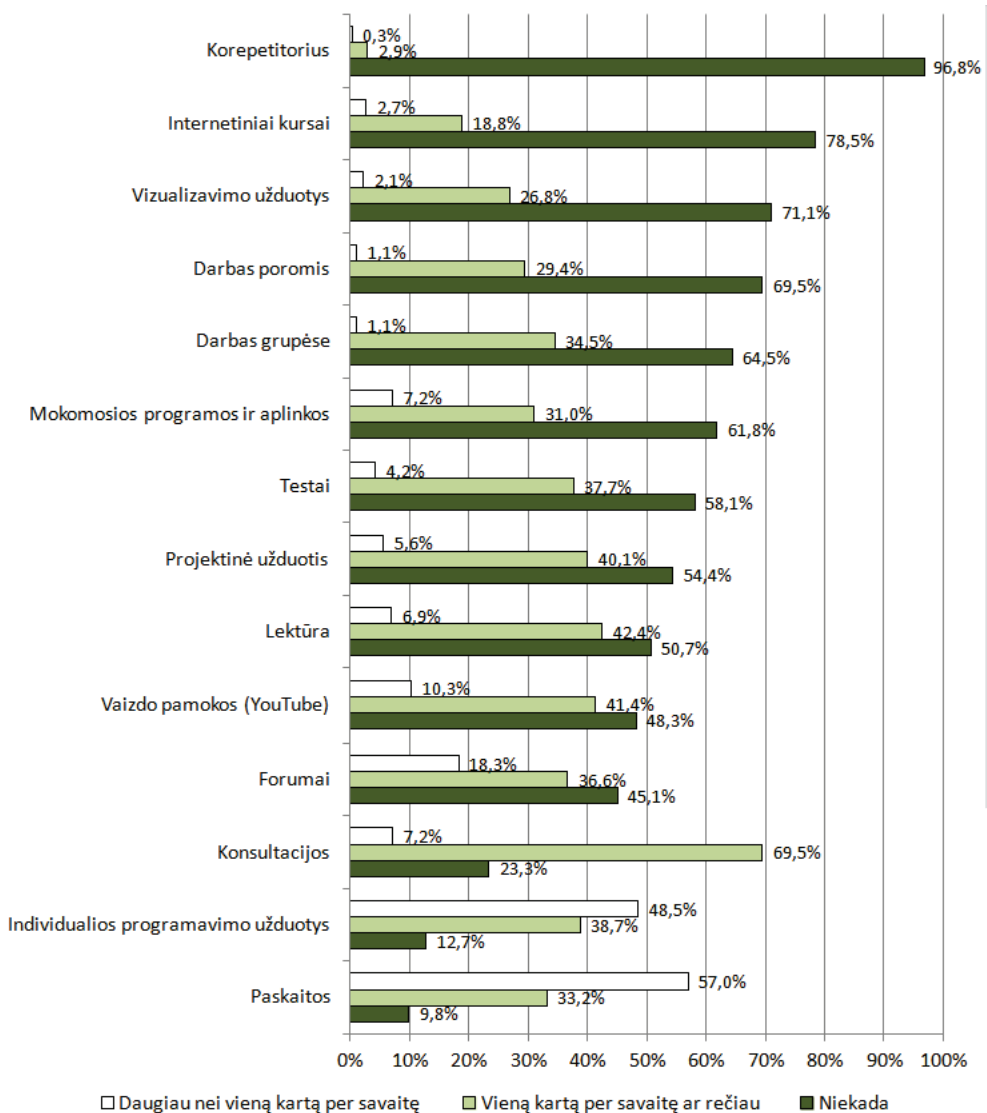


20 pav. Programavimo mokymosi sunkumų ir patirties ryšys

Respondentai dalyvavo apklausoje jau išklausę ĮPK aukštojoje mokykloje. Siekiant išsiaiškinti, kaip jie vertina išklausyto kurso poveikį programavimo gebėjimams, buvo klausiama, ar išklausytas ĮPK padėjo pagerinti programavimo gebėjimus. Išklausytą ĮPK teigiamai vertina 70,8 proc. studentų. Lyginant studentų atsakymus pagal aukštąsias mokyklas ir studijų kryptis reikšmingo skirtumo nenustatyta; ryšys tarp turimos programavimo patirties ir išklausyto ĮPK vertinimo taip pat nenustatytas.

Mokymosi metodų taikymas. Aiškintasi, kokie studijų metodai ir kaip dažnai jie taikomi studijuojant ĮPK kursą. Nustatyta (žr. 21 paveikslą), kad dažniausiai taikyta: paskaitos (90,2 proc.), individualios programavimo užduotys (87,2 proc.) ir konsul-

tacijos (76,7 proc.). Daugiau kaip pusė apklaustųjų mokymdamiesi naudojasi forumais (54,9 proc.) ir vaizdo pamokomis (51,7 proc.), o kiek mažiau kaip pusė mokymdamiesi skaito knygas (49,3 proc.). Bendradarbiavimą skatinančius darbo grupėse (35,6 proc.) ar porose (30,5 proc.) metodus respondentai taikė rečiau negu mokomąsias programas (38,2 proc.) ar testus (41,9 proc.). Kiti studijų metodai taikyti dar rečiau. Rečiausiai studijuojant ĮPK naudotasi korepetitoriaus pagalba (3,2 proc.).



21 pav. Studijuojant ĮPK taikyti studijų metodai

Analizuoti ir mokymosi metodų taikymo skirtumai pagal aukštąsias mokyklas. Kaip jau minėta, mokantis programuoti dažniausiai taikomas studijų metodas yra paskaita. Universitetų ir kolegijų studentų atsakymai skyrėsi nurodant šio metodo taikymo dažnumą. Kelis kartus per savaitę paskaitos dažniau skaitomos kolegijose (80,3 proc.) nei universitetuose (32,6 proc.), o kartą per savaitę ar rečiau – dažniau universitetuose (60,3 proc.) nei kolegijose (7,3 proc.). Panašiai nuomonės pasiskirstė ir tiriant individualias programavimo užduotis. Vertinant šių metodų ryšį pagal aukštąsias mokyklas nustatyti statistiškai reikšmingi skirtumai (žr. 24 lentelę).

Individualias konsultacijas su dėstytoju labai dažnai renkasi ir universitetų (77,9 proc.), ir kolegijų (77,7 proc.) studentai. Šiuo atveju statistiškai reikšmingo skirtumo nenustatyta.

Internetas suteikia galimybių mokytis tiesiogiai ir nuotoliniu būdu. Programavimui mokytis taip pat gali būti pasitelkiamos interneto platybės, todėl respondentų buvo klausama, kaip dažnai jie naudojami forumais, internetiniais kursais ar vaizdo įrašų saugyklomis. Nustatyta, kad forumais dažniau naudojami universitetų (65,2 proc.) nei kolegijų (45,1 proc.) studentai. Internetiniai kursai nebuvo populiarūs, juose mokėsi maždaug kas penktas universitetų (23,4 proc.) ir kolegijų (19,2 proc.) studentas. Universitetų (62,5 proc.) studentai vaizdo pamokas vaizdo įrašų saugyklose žiūrėjo dažniau negu kolegijų (41,4 proc.) studentai. Nustatyti statistiškai reikšmingi forumų ir vaizdo pamokų naudojimosi skirtumai (žr. 24 lentelę).

Studentų mokymosi įpročiai rodo, kad jie mažiau naudojami spausdintiniais mokymosi literatūros šaltiniais. Programavimui mokytis knygomis visai nesinaudoja du kartus daugiau kolegijų (67,4 proc.) negu universitetų (33,2 proc.) studentų. Analizuojant šiuos duomenis pagal aukštąsias mokyklas, nustatytas statistiškai reikšmingas skirtumas (žr. 24 lentelę).

Darbo grupėse metodai mokantis programuoti taikomi panašiai tiek universitetuose (32,6 proc.), tiek ir kolegijose (38,3 proc.). Tačiau daugiau kaip pusė respondentų (64,5 proc.) darbo grupėse metodų visai nebuvo išbandę. Statistiškai reikšmingo skirtumo, dirbant grupėse, tarp aukštųjų mokyklų nenustatyta. Porose dirbo dar mažiau tiriamųjų (30,5 proc.). Lyginant duomenis pagal aukštąsias mokyklas nustatytas silpnas statistinis ryšys (žr. 24 lentelę), rodantis, kad universitetuose (36,9 proc.) šis metodas taikomas šiek tiek dažniau nei kolegijose (24,3 proc.).

Vizualizavimo užduotys, padedančios suvokti programos vykdymo eigą (programos veikimo sekimas) arba identifikuoti žinių spragas (minčių žemėlapis), universitetuose (38 proc.) naudojamos dažniau nei kolegijose (18,6 proc.). Nustatytas statistiškai reikšmingas skirtumas (žr. 24 lentelę).

Tiriant statistinius ryšius tarp taikytų mokymosi metodų ir aukštųjų mokyklų nustatytas statistiškai reikšmingas ryšys šių mokymosi metodų ir būdų: paskaitų, individualių programavimo užduočių, vizualizavimo užduočių, darbo porose, vaizdo pamokų, forumų, mokomųjų programų ir aplinkų, lektūros. Statistiškai reikšmingas kitų mokymosi būdų (darbo grupėse, konsultacijų, korepetitoriaus pagalbos, internetinių kursų ir testų) skirtumas pagal aukštąsias mokyklas nenustatytas.

Analizuojant taikytus mokymosi metodus pagal studijų kryptis statistiškai reikšmingų skirtumų nenustatyta. Remiantis turimais duomenimis nenustatyti statistiniai ryšiai tarp taikytų studijų metodų ir išklaustyto ĮPK kurso poveikio programavimo gebėjimams vertinimų.

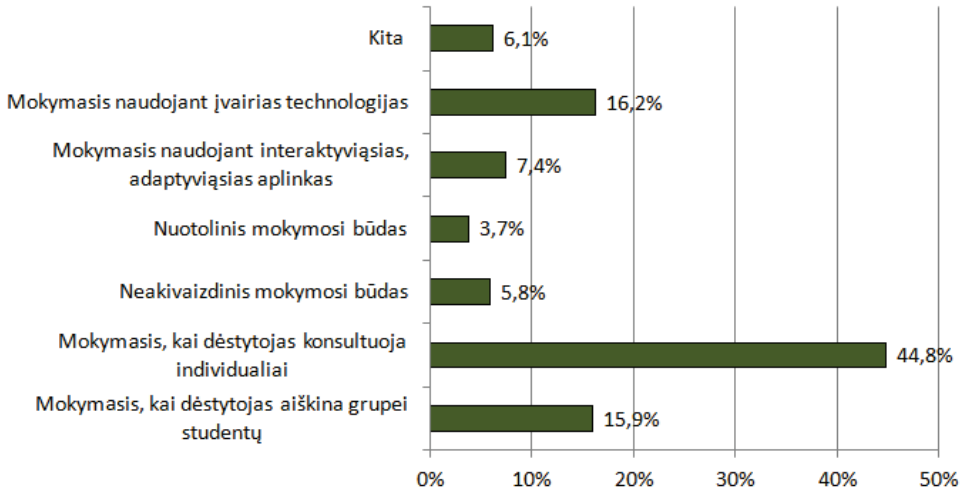
24 lentelė

ĮPK dalyko studijų metu taikytų mokymosi metodų dažnis pagal aukštąsias mokyklas

Studijų metodai	Aukštosios mokyklos	Studijų metodo taikymo dažnis						Ryšys pagal aukštąsias mokyklas
		Niekada		Kartą per savaitę ar rečiau		Dažniau nei kartą per savaitę		
		Skaičius	Proc.	Skaičius	Proc.	Skaičius	Proc.	
Paskaitos	Kolegijos	24	12,4	14	7,3	155	80,3	$\chi^2 = 120,4;$ df = 2; p < 0,001
	Universitetai	13	7,1	111	60,3	60	32,6	
Individualios programavimo užduotys	Kolegijos	32	16,6	38	19,7	123	63,7	$\chi^2 = 60,4;$ df = 2; p < 0,001
	Universitetai	16	8,7	108	58,7	60	32,6	
Darbas grupėse	Kolegijos	119	61,7	73	37,8	1	0,5	$\chi^2 = 2,86;$ df = 2; p > 0,05
	Universitetai	124	67,4	57	31,0	3	1,6	
Darbas porose	Kolegijos	146	75,6	46	23,8	1	0,5	$\chi^2 = 7,47;$ df = 2; p < 0,02
	Universitetai	116	63,0	65	35,3	3	1,6	
Vizualizavimo užduotys	Kolegijos	157	81,3	34	17,6	2	1,0	$\chi^2 = 20,48;$ df = 2; p < 0,001
	Universitetai	111	62,1	67	35,2	6	2,8	
Projektinės užduotys	Kolegijos	127	65,8	66	34,2	0	0	$\chi^2 = 34,91;$ df = 2; p < 0,001
	Universitetai	78	42,4	85	46,2	21	11,4	
Vaizdo pamokos (YouTube)	Kolegijos	113	58,5	72	37,3	8	4,1	$\chi^2 = 24,92;$ df = 2; p < 0,001
	Universitetai	69	37,5	84	45,7	31	16,8	

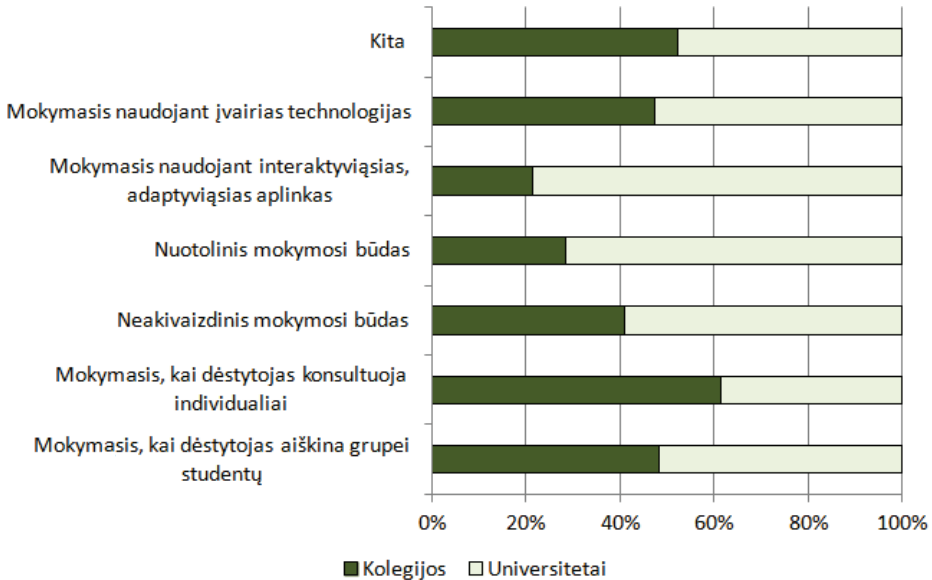
Studijų me- todai	Aukštosios mokyklos	Studijų metodo taikymo dažnis						Ryšys pagal aukštąsias mokyklas
		Niekada		Kartą per savaitę ar rečiau		Dažniau nei kartą per sa- vaitę		
		Skaičius	Proc.	Skaičius	Proc.	Skaičius	Proc.	
Forumai	Kolegijos	106	54,9	59	30,6	28	14,5	$\chi^2 = 15,52$; df = 2; p < 0,001
	Universitetai	64	34,8	79	42,9	41	22,3	
Mokomosios programos ir aplinkos	Kolegijos	128	66,3	59	30,6	6	3,1	$\chi^2 = 10,4$; df = 2; p < 0,006
	Universitetai	105	57,1	58	31,5	21	11,4	
Korepetito- rius	Kolegijos	188	97,4	4	2,1	1	0,5	$\chi^2 = 1,94$; df = 2; p > 0,05
	Universitetai	177	96,2	7	3,8	0	0	
Lektūra	Kolegijos	130	67,4	58	30,1	5	2,6	$\chi^2 = 46,69$; df = 2; p < 0,001
	Universitetai	61	33,2	102	55,4	21	11,4	
Internetiniai kursai	Kolegijos	156	80,8	34	17,6	3	1,6	$\chi^2 = 2,38$; df = 2; p > 0,05
	Universitetai	140	76,6	37	20,0	7	3,4	
Testai	Kolegijos	122	63,2	63	32,6	8	4,1	$\chi^2 = 4,44$; df = 2; p > 0,05
	Universitetai	97	39,3	79	56,6	8	4,1	
Konsultacijos	Kolegijos	43	22,3	134	69,4	16	8,3	$\chi^2 = 0,89$; df = 2; p > 0,05
	Universitetai	45	22,1	128	71,7	11	6,2	

Apklausiant studentus buvo prašoma nurodyti programavimo mokymosi būdus, kurie, respondentų manymu, tinkamiausi ir naudingiausi mokantis programuoti. Pastebėta, kad tinkamiausiu didžioji dalis respondentų nurodė individualių konsultacijų metodą (44,8 proc.). Tik šeštadalis (16,2 proc.) respondentų programavimui mokyti pasirinktų įvairias technologijas. Kiek mažiau kaip šeštadaliui (15,9 proc.) tiriamųjų tinkamiausias įprastinis mokymasis, kai dėstytojas aiškina grupei studentų (žr. 22 paveikslą).



22 pav. Respondentų pageidaujami mokymosi būdai

Analizuojant pageidaujamus mokymosi būdus pagal aukštąsias mokyklas nustatytas reikšmingas skirtumas ($\chi^2 = 46,69$; $df = 2$; $p < 0,001$), rodantis, kad kolegijų studentai mokydami dažniau pageidauja paramos. Universitetų studentai dažniau renka savarankiško mokymosi metodus (žr. 23 paveikslą).



23 pav. Respondentų pageidaujami mokymosi būdai kolegijose ir universitetuose

Apibendrinant anketinio tyrimo rezultatus galima teigti, kad nevienodas pradinis pasirengimas programavimo studijoms būdingesnis kolegijų studentams. Programavimo gebėjimams ugdytis Lietuvos aukštosiose mokyklose taikomi įvairūs studijų metodai, tačiau kolegijų studentai mažiau dirba savarankiškai ir pageidauja daugiau dėstytojo konsultacijų. Daugelio universitetų ir kolegijų studentų darbo grupėse metodai netaikomi (statistiškai reikšmingo skirtumo nenustatyta), nors dalykų aprašų analizė rodo, kad kolegijose šie metodai turėtų būti taikomi dažniau.

3.1.3. Kokybinis tyrimas „Įvadinio programavimo dėstyto strategijos“

Atsižvelgiant į tai, kad ĮPK aprašų analizės ir studentų anketinės apklausos duomenys studijų metodų taikymo klausimu šiek tiek prieštaringi, suorganizuotas dėstytojų interviu. Jo metu buvo siekiama išsiaiškinti dėstytojų požiūrį apie naudojamas įvadinio programavimo mokymosi strategijas: kokie mokymosi metodai ar specializuotos kompiuterinės mokymosi priemonės naudojamos kursui dėstyti? Kitas aspektas, kurį mėginta išsiaiškinti, – ar programavimo mokymasis individualizuojamas diferencijuojant studentus arba mokymosi turinį.

Analizuojant interviu duomenis nustatyta, kad dažniausiai studijos organizuojamos pagal įprastą schemą: per savaitę skaitoma 2 akademiniai valandų paskaita ir vykdomos 2–4 akademiniai valandų pratybos. Informantų buvo prašoma įvertinti, kiek tokia studijų schema pasiteisina mokant programuoti. Dauguma tvirtino, kad būtent mokantis programuoti tokia praktika nėra gera: „<...> paskaitų skaitymas, kai studentai sėdi ir klausosi, mokant programavimo, nėra gera praktika“ (In-3)¹¹; „iš tikrųjų manyčiau, kad dėstytojui yra patogiau, kai tiek paskaitos, tiek pratybos organizuojamos kompiuterinėje auditorijoje, kur aš galiu pasakoti ir rodyti, ir paraginti studentus kažką tai praktiškai išbandyti“ (In-2). Pasakodami apie savo skaitomas paskaitas informantai nurodė ir originalius paskaitų laiko išnaudojimo atvejus: „Tai per teorines (paskaitas – V. J.) aš stengiuosi per daug jų neapkrauti tom sąvokom, o duodu ant lapo tokių užduotėlių, pakalbu kokių 15 minučių ir likusį laiką bandau duoti jiems spręsti“ (In-4); „<...> savo praktikoje mėgstu įdarbinti studentus. Kadangi turiu savo kurso teorinę medžiagą parengęs Moodle terpėje, tai studentams būna duodama užduotis perskaityti teoriją, o paskaitą tada organizuoju kaip diskusiją, atsakymą į iškilusius klausimus“ (In-5); „Norint šiek tiek suaktyvinti, dažnai užduodu probleminius klausimus“

¹¹ Informantų pasisakymai cituojant neredaguoti.

mus, organizuoju diskusiją, ar darbą grupėmis, piešiame algoritmų blokines schemas ar minčių žemėlapius <...>“ (In-3). Pratybos ar laboratoriniai darbai organizuojami irgi panašiai: „Per pratybas mes taip dirbam, praktiškai yra visa pratybų medžiaga parengta iš anksto, pirmiausia studentai atlieka užduotis, kad susipažintų ir įsisavintų naują medžiagą, o po to jie jau atlieka kiekvienas individualiai laboratorinius darbus“ (In-1); „Praktika, tai jau užduotys daromos prie kompo ir turi apsiginti būtinai“ (In-4); „Jeigu yra pratybos, toliau yra pateikiami uždaviniai ir tie uždaviniai yra realizuojami <...>“ (In-2); „Visai darom, pradėjus naują temą daugiau sprendžiame kartu <...> po to pabandom dirbti grupėmis, o galiausiai (studentai – V. J.) atlieka individualias užduotis“ (In-3).

Kalbėdami apie naudojamą kompiuterines mokymosi priemones informantai dažniausiai minėjo integruotas programavimo aplinkas bei virtualiąją mokymosi aplinką (Moodle) ir jos suteikiamas mokymosi organizavimo priemones: forumus, tinklaraščius, vikius, žodynėlius: „Užduotį jie analizuoja kaip vikį, o aš matau, kas ir kiek dirba, taip pat forumuose dalyvauja, diskutuoja ir aptarinėja“ (In-5). Informantas In-3 pastebėjo, kad „forumai ir pokalbiai Moodle aplinkoje jiems nepatinka, jie persikėlė šitas veiklas į Facebook, turi savo uždara grupę, ten ir dalijasi visomis gėrybėmis“. Vienas iš informantų (In-4) algoritmams ir programavimo sąvokoms aiškinti teigė pasitelkiantis internetines vizualiojo programavimo aplinkas (Blockly, Turtle), leidžiančias dėlioti programėles iš įvairių blokų ir čia pat stebėti jų veikimą.

Klausiami apie vertinimo metodus, visi informantai nurodė, kad kurso studijos baigiamos egzaminu, tačiau taikomas kaupiamojo balo principas: dalį galutinio vertinimo studentai sukaučia semestro metu, o kitą dalį gauna užbaigę kurso studijas ir išlaikę egzaminą. Semestro metu atliekami ir ginami individualūs laboratoriniai darbai, rašomi kontroliniai darbai, laikomi testai, atliekamos žodinės apklausos pasikaitų metu ir pan. Naudojantieji virtualiąsias mokymosi aplinkas teigė sudarantys sąlygas studentams pasitikrinti savo žinias, atliekant paruoštus savikontrolės testus su automatinio grįžtamuoju ryšiu.

Analizuojant informantų atsakymus apie mokymosi individualizavimą nustatytas vienas atvejis, kai studentai diferencijuojami pagal gebėjimus: „Yra tas vadinamasis A lygis ir S lygis. Į S lygį eina tie, kurie jau yra programavę, ir jie jau truputį kitaip dirba, o pradėdantieji vėl truputį kitaip dirba. Skiriasi jų visų darbų sudėtingumas. Jiems (A lygio – V. J.) studentams yra šiek tiek lengvesnės užduotys. S lygis gali rinktis, jie ir grupinius projektus daro. O dėl A lygio, tai mes jau stengiamės, kad jie (studentai – V. J.) įgytų tuos tvirtus pradinius programavimo įgūdžius“ (In-1). Kiti informantai teigė, kad individualizavimas „kažkiek tai vyksta, bet tik tokiam intuityviame lygmenyje, ir matai,

jeigu žmogus klausia, kai kažkas jam neaišku, tai gali sugalvoti kažkokių pavyzdėlių, sakykim, tiesiog pagal jo lygį“ (In-4); „Bandau orientuoti pačias užduotis į studentus, populiarinant programavimą, kad kuo labiau būtų taikomojo pobūdžio, nes kitaip nėra kažkokiai tokio užsidegimo“ (In-2); „Taip, Moodle terpėje turiu uždavinukų su žvaigždute, kartais tie, kuriems viskas aišku, sprendžia juos, o aš tuo tarpu konsultuoju tuos, kuriems sunkiau sekasi“ (In-5).

Interviu nebuvo griežtai struktūruotas, bet keli informantai nurodė pradedančiųjų patiriamus sunkumus mokantis programuoti ir galimas to priežastis. Informantas In-4 nurodė, kad sunkiausia, kai „studentai padaro programose klaidų, pasimeta ir nežino, kaip jas ištaisyti“, o informantas In-2 nurodė galimą tokios ugdymo situacijos priežastį: „Pirmiausia jie (studentai – V. J.) negalvoja, o bando „goglinti“ (ieškoti internete – V. J.), jeigu randa kažką panašaus, jie būtinai pasiima ir bando pritaikyti savo programoje tuos gabaliukus arba visą programą. Dėl to jie nekuria savo sprendimo, o bando adaptuoti jau esantį <...>. Didelė dalis pavyzdžių randama su angliškais komentarais arba su angliška notacija, ir tada dalis programos lieka tiesiog nežinoma. Kita problema kyla dėl rasto pavyzdžio lygmens neatitikimo, tiesiog pavyzdys rašytas aukštesnio lygio programuotojo, tarkime, C kalboje naudojame nullterminated strings, o jie rado uždavinuką su string tipo naudojimu. Ir tuomet atitinkamai juk kitas funkcionalumas ir kitos tvarkymo galimybės. Ir studentas tuomet pasimeta, kodėl dėstytojas sako vienaip, o aš radau kitaip. Ir kaip tą kitaip pritraukti prie turimo uždavinuko“. Informantas In-1 akcentavo informacijos paieškos gebėjimų trūkumą: „Jie (studentai – V. J.) iš tikrųjų pripratę labai greitai gauti pagalbą. Žodžiu, jie labai nemėgsta knistis ir surasti patys. Jie nemoka tinkamai ieškoti informacijos internete“ (In-1). Užsiminta, kad pirmojo kurso studentų mokyklinės matematikos žinios prastos: „Iš mokyklos dabartiniai studentai ateina su didžiulėmis spragom, kalbant apie žodinius uždavinius. Kadangi dauguma programavimo uždavinių yra žodiniai <...>, man atrodo, kad mokykloje mokiniams sunkiausiai sekasi ir labiausiai nemėgstami mokinių būtent žodiniai uždaviniai. <...> ir aš manau, kad tai yra bendra tendencija“ (In-2). Informantas In-3 pradedančiųjų patiriamus sunkumus susiejo su programavimo užduoties sprendimo etapais: „Kai kuriems sunkus užduoties apmąstymo ir sprendimo kelio sugalvojimo etapas, <...> na, o kita rūšis sunkumų – tai sugalvoto algoritmo užrašymas programavimo kalbos sakiniiais, na, turi praeiti tam tikras laiko tarpas, kad ateitų supratimas, iš pradžių jie nemąsto programavimo sąvokomis, bet besipraktikuojant tai įveikiama, svarbu – daug ir kruopščiai dirbti“ (In-3).

Verta paminėti vieno iš informantų (In-2) nurodytus dabartinės studentų kartos mąstymo ypatumus: „Idealus variantas studentui yra šabloniniai uždaviniai. Bet vėlgi,

manųčiau, kad ta tendencija ateina iš mokyklos, <...> jis (studentas – V. J.) išmoksta spręsti eilę šabloninių uždavinių, ir mąstymas tampa orientuotas į tam tikrų šablonų atpažinimą. Jis nebando kažką tai interpretuoti, kažką tai pasvajoti savo, o bando ieškoti tam tikro šablono. Jeigu neranda šablono – pasimeta. Tas persilaužimas įvyksta pakankamai sunkiai. Keliai yra du: arba jis pradeda formuotis naujus šablonus, arba jis bando visgi pradėti mąstyti nestandartiškai“ (In-2).

Apibendrinant informantų pasisakymus galima išskirti pradedantiesiems trūkstamus gebėjimus ir asmenines savybes:

- uždavinio ir jo sprendimo apmąstymo (refleksijos),
- turimų žinių susiejimo su problema,
- informacijos paieškos ir analizės,
- atkaklumo ir kruopštumo sprendžiant uždavinius,
- pagrindinių matematinių ir loginių žinių taikymo.

Pastebėta, kad interviu dalyviai išskyrė tam tikrą programavimo mokymosi problemų grupę, kuri mokslinėje literatūroje įvardijama kaip besimokančiųjų patiriami „sunkumai, susiję su besimokančiųjų gebėjimais ir nuostatomis“ (Gomes, Carmo ir kt., 2006).

Informantų buvo prašoma nurodyti, kas gi yra svarbiausia dėstant ĮPK. Jų nuomonės šiuo klausimu šiek tiek skyrėsi: „<...> prioritetas turi būti teikiamas pagrindinėms struktūroms išaiškinti. Jeigu mes kalbam apie struktūrinį programavimą, tai būtų sąlyginiai sakiniai, ciklai, t. y. pagrindiniai programavimo konceptai, kurie naudojami būtent tam tikroje paradigmoje“ (In-2); „<...> svarbiau loginis mąstymas ir algoritmai“ (In-5); „<...> labiau akcentuoju vis tik gebėjimą logiškai galvoti, analizuoti užduotį, rengti sprendimo algoritmą“ (In-3); „Vieni akcentuoja programavimo kalbų svarbą, kiti labiau akcentuoja duomenų struktūrų svarbą, dar kiti – algoritmų, o aš tai manau, kad algoritmas yra pirmiausia“ (In-1). Taigi, dėstytojų nuomone, labai svarbus yra loginis ir algoritminis mąstymas.

3.1.4. Žvalgomojo tyrimo rezultatų apibendrinimas

Atlikta įvadinio programavimo kursų aprašų analizė leidžia teigti, kad:

- programavimo studijos Lietuvos aukštosiose mokyklose dažniausiai pradedamos nuo procedūrinio programavimo, o įvadinis programavimo kursas apsiriboja viena procedūrinio programavimo kalba (dažniausiai C arba C++). Universitetuose dažniau nei kolegijose baigiant įvadinio programavimo kurso

studijas skiriama laiko ir objekcinio programavimo sąvokoms. Kitų pradedančiųjų programavimo mokymosi paradigmu (kai pradedama nuo loginio programavimo; kai pradedama nuo algoritmu) sutinkama tik universitetuose. Kolegijose ĮPK dėstyti pasirenkamos įvairesnės programavimo kalbos;

- tarp ĮPK tikslų dažniausiai numatomas tik programavimo konceptų supratimas ir programavimo kalbos sintaksės žinių suteikimas, o programavimo užduočių analizės, problemų sprendimo ir programinių fragmentų derinimo gebėjimų ugdymas minimas rečiau;
- mokymosi metodų įvairovė kolegijų ĮPK aprašuose didesnė negu universitetų, tačiau aktyvaus darbo metodų dažnis nedidelis tiek kolegijų, tiek universitetų ĮPK aprašuose.

Atlikta studentų anketinės apklausos duomenų analizė leidžia teigti, kad:

- programavimo studijas Lietuvos aukštosiose mokyklose be jokios išankstinės programavimo patirties pasirenka kiek daugiau kaip ketvirtadalis pirmakursių. Nustatytas statistiškai reikšmingas skirtumas rodo, kad dažniau be programavimo patirties pasirenkamos studijos kolegijose, informatikos inžinerijos ir informatikos kryptių studijų programose; beveik trečdalis į kolegijas susirinkusių pirmojo kurso studentų nėra mokęsi programavimo pradmenų bendrojo ugdymo mokykloje;
- programų sistemų studijas dažniau renkasi labiau motyvuoti studentai, kuriems programuoti patinka. Kiek daugiau kaip pusė ĮPK išklausių studentų programavimo mokymąsi vertina kaip labai sunkų. Taip manančiųjų kolegijose daug daugiau;
- pradedančiųjų patiriami programavimo mokymosi sunkumai susiję daugiau su motyvacijos ir savarankiško darbo trūkumu. Patiriamus programavimo mokymosi sunkumus panašiai nurodo ir kolegijų, ir universitetų studentai. Nuomonės neišsiskiria nei pagal studijų kryptis, nei pagal turimą programavimo patirtį;
- kolegijų ir universitetų studentų nuomonės šiek tiek išsiskiria vertinant programavimo gebėjimams ugdyti taikytus studijų metodus ir būdus. Kolegijų studentai rečiau renkasi savarankiško mokymosi metodus, pageidauja daugiau dėstytojo konsultacijų. Bendradarbiavimą skatinančių mokymosi metodų taikymo dažnis kolegijose ir universitetuose skiriasi labai mažai.

Dėstytojų interviu duomenų analizė parodė, kad:

- pradedančiųjų programavimo gebėjimams ugdyti dėstytojai taiko įvairių mokymosi ir vertinimo metodų, pasitaiko ir gana originalių, ir inovatyvių mokymosi organizavimo būdų;

- informantų nuomone, ugdant pradedančiųjų programavimo gebėjimus paskaitų skaitymas nėra veiksmingas mokymosi būdas. Aukštosioms mokykloms vertėtų apsvastyti kitokią pradedančiųjų programavimo gebėjimų ugdymo modelį, kai nauja mokymosi medžiaga pateikiama pasitelkiant informacines komunikacines technologijas, o kontaktinis laikas naudojamas praktiniams gebėjimams ugdyti ir konsultacijoms;
- ugdant pradedančiųjų programavimo gebėjimus yra mokymosi individualizavimo apraiškų. Dažniau mokymasis diferencijuojamas pagal besimokančiųjų turimus programavimo gebėjimus, rečiau – pagal ugdymosi turinį. Nenustatyta, kad mokymasis individualizuojamas atsižvelgiant į besimokančiųjų mokymosi stilių, nors 3 informantai teigė žinantys apie tokią galimybę;
- apibūdindami besimokančiųjų patiriamus mokymosi sunkumus kurso dėstytojai išskiria tik tam tikrą programavimo mokymosi sunkumų, susijusių su besimokančiųjų gebėjimais ir nuostatomis, grupę.

Apibendrinant žvalgomojo tyrimo rezultatus galima teigti, kad universitetų IPK aprašuose vyrauja tradiciniai studijų metodai, o kolegijų – aktyvaus darbo studijų metodai (darbas grupėse, diskusijos, minčių lietus, probleminis dėstymas ir kt.). Galima manyti, kad studijų dokumentuose pateikiama informacija ne visuomet atitinka tai, kaip praktiškai ugdomi pradedančiųjų programavimo gebėjimai, mat, studentų apklausos duomenimis, nėra didelio skirtumo tarp kolegijose ir universitetuose taikomų mokymosi metodų. Statistiškai reikšmingi skirtumai nustatyti šiems studijų metodams: paskaitai, individualiam darbui, lektūrai, vizualizavimo užduotims. Tačiau šis skirtumas parodė tik skirtingas paskaitų ir praktiniams gebėjimams ugdyti numatyto laiko proporcijas universitetuose (daugiau paskaitų) ir kolegijose (daugiau praktinio darbo). Kitus mokymosi metodus (darbą grupėmis, internetinius kursus, diskusijas ir kt.) mokantis programuoti panašiai taiko ir kolegijų, ir universitetų studentai. Patikimesni šiuo atveju anketinio tyrimo rezultatai. Taigi galima teigti, kad aukštosiose mokyklose (universitetuose ir kolegijose) iš esmės vyrauja mokymo paradigmą atitinkantis studijų modelis: studentai klauso paskaitų ir sprendžia individualias programavimo užduotis, konsultuojasi su dėstytojais ir bendramoksliais.

Atliekant disertacinį tyrimą buvo sukurtas MSMMMD modelis ir numatyti jo panaudojimo atvejai (žr. 2.2 skyrių). Norint taikyti modelį būtina informuoti studentus, skatinti įsitraukti, suteikti galimybę rinktis kuo daugiau mokymosi metodų. Vadinasi, norint taikyti modelį programavimo gebėjimams ugdyti, būtina didinti ir programavimui mokytis taikomų studijų metodų įvairovę. O žvalgomojo tyrimo

rezultatai rodo, kad ugdant programavimo gebėjimus kai kurie mokymosi metodai taikomi labai retai arba visai netaikomi.

3.2. MOKYMOŠI STILIŲ IR MOKYMOŠI METODŲ DERMĖS MODELIO TYRIMAS

3.2.1. Mokymosi metodų vertinimai

Pradedančiųjų programavimo gebėjimams ugdytis naudota disertacinio tyrimo metu sukurta įvadinio programavimo kurso programa (žr. 6 priedą). Vykdamt ĮPK programą panaudota tam tikra dalis pradedančiųjų programavimo gebėjimams ugdytis taikytų mokymosi metodų, jų sąrašas pateiktas 9 priede.

Po kiekvieno užsiėmimo, organizuoto pagal iš anksto numatytą planą, studentai buvo apklausiami, prašoma įvertinti užsiėmimo metu taikytus mokymosi metodus. Atliekant tyrimą sukauptas skirtingas taikytų mokymosi metodų vertinimų skaičius (žr. 25 lentelę). Renkant duomenis įtakos turėjo studentų lankomumas ir aktyvumas vertinimų metu – ne visi tyrimo dalyviai pateikdavo išsamių vertinimų. Todėl tyrinėtus metodus – durstinį, varžybas, minčių lietu, problemų sprendimą, sąvokų žemėlapi – vertino labai mažai studentų. Studentų požiūriui apie taikytus mokymosi metodus atskleisti parengtos ir panaudotos kelių rūšių vertinimo formos su skirtingomis vertinimo skalėmis. Tad, apibendrinant studentų požiūrį, vertinimai buvo koduojami taip:

- 1 – mokymosi metodas vertinamas neigiamai (netinka, nepatinka, nelabai patinka);
- 2 – mokymosi metodas vertinamas vidutiniškai (neblogai, nei patinka, nei nepatinka);
- 3 – mokymosi metodas vertinamas teigiamai (tinka, patinka, labai patinka).

Mokymosi metodų vertinimai*

Kodas	Mokymosi metodai	Vertinimai			Bendras vertinimų kiekis
		1	2	3	
MM01	Algoritmų ir programų vizualizavimas	15	14	11	40
MM04	Darbas grupėse bendradarbiaujant	9	14	29	52
MM05	Problemų sprendimas	2	3	2	7
MM06	Demonstravimas	2	16	24	42
MM07	Diskusija	6	6	19	31
MM08	Programavimas porose	4	12	22	38
MM09	Durstinys	2	1	4	7
MM10	Programos teksto skaitymas ir veikimo sekimas	5	13	27	45
MM11	Individualus darbas	12	21	17	50
MM12	Minčių lietus	2	9	3	14
MM13	Paskaita	25	29	9	63
MM15	Pristatymai	6	17	22	45
MM16	Savikontrolės testas	8	11	16	35
MM17	Sąvokų žemėlapis	5	1	1	7
MM18	Vaizdo pamoka	6	2	1	9
MM21	Varžybos	9	2	3	14
	Iš viso	118	171	210	499

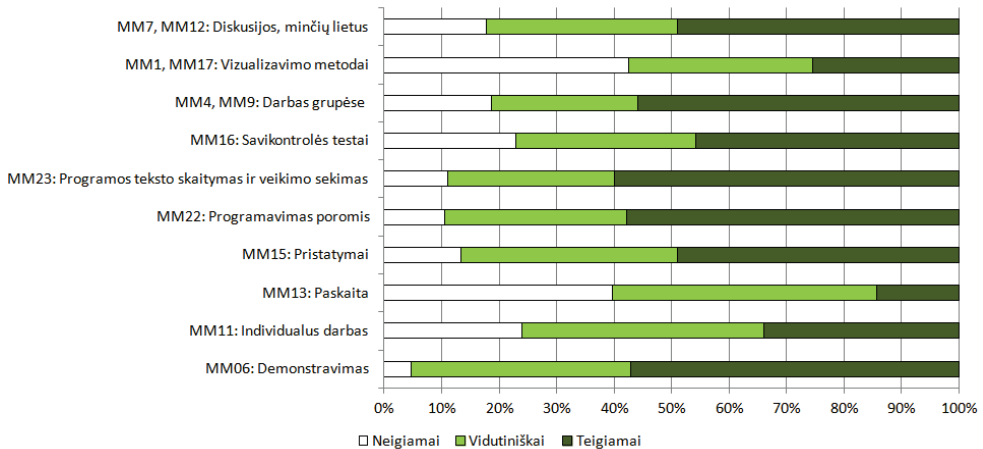
* Mokymosi metodų vertinimai: 1 – neigiamas (netinka, nepatinka, nelabai patinka); 2 – vidutinis (neblogai, nei patinka, nei nepatinka); 3 – teigiamas (tinka, patinka, labai patinka).

Apibendrinant ir vertinant studentų išsakytą požiūrį apie taikytus mokymosi metodus reikia atsižvelgti ir į veiklos tyrimo keliamus apribojimus. Tyrime dalyvavo palyginti mažas tiriamųjų skaičius – po 7 kiekviename veiklos tyrimo cikle. Taip pat skyrėsi mokymosi metodų taikymo dažnumas (pvz.: durstinio metodas taikytas 3 kartus, demonstravimas – 30 kartų). Kai kuriais atvejais buvo susiduriama su menka respondentų mokymosi proceso vertinimo ir įsivertinimo patirtimi arba nenoru vertinti. Visi išvardyti ribojimai neleidžia formuluoti kiekybiškai pagrįstų išvadų, tačiau sukauptų mokymosi metodų vertinimų kiekybinė analizė vis dėlto atskleidžia tam tikras tendencijas.

Analizuojant studentų požiūrį į taikytus mokymosi metodus pastebėta, kad vykdytos veiklos beveik du kartus dažniau įvertintos teigiamai negu neigiamai. O vidutinių vertinimų kiekis sudaro kiek daugiau kaip trečdali visų vertinimų. Paprastai labiau teigiamai negu neigiamai studentai vertino diskusijų (61,3 proc.), programų teksto skaitymo ir veikimo sekimo (60 proc.), programavimo porose (57,5 proc.), demonstravimo (57,1 proc.), durstinio (57,1 proc.), darbo grupėse (55,8 proc.) ir pristatymų (48,9 proc.) metodus. Savikontrolės testai šiek tiek dažniau vertinti teigiamai

(45,7 proc.) negu vidutiniškai (31,4 proc.). O dažniau vidutiniškai (42 proc.) negu teigiamai (34 proc.) vertintas individualus darbas. Daug dažniau neigiamai vertintos vaizdo pamokos, varžybos ir sąvokų žemėlapis, tačiau šie metodai taikyti tik keletą kartų ir jiems sukauptas palyginti mažas vertinimų kiekis. Paskaitos taip pat dažniau vertintos neigiamai (39,7 proc.) negu teigiamai (14,3 proc.), o algoritmų ir programų vizualizavimas neigiamai įvertintas 15 kartų iš 40 (37,5 proc.).

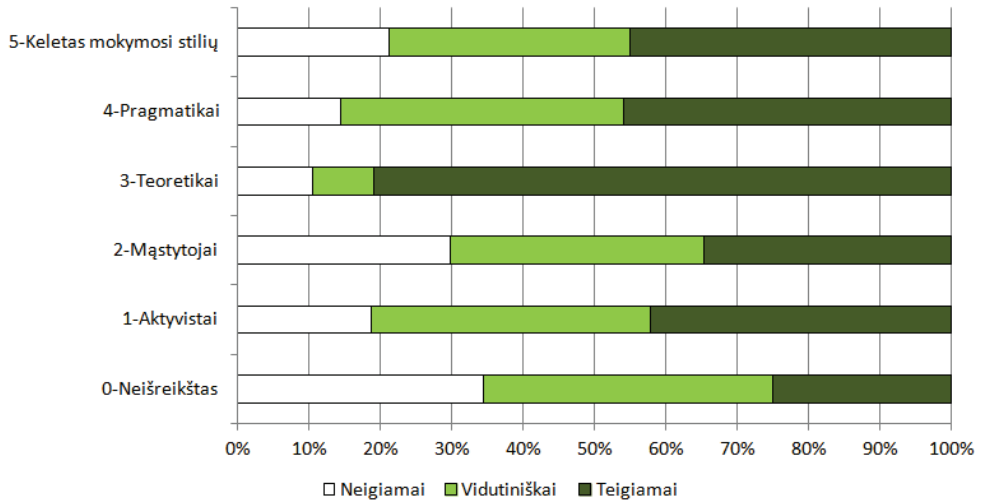
Dažniau teigiamai (žr. 24 paveikslą) studentai vertino tuos metodus, kurie labiau atitinka dalyko poreikius (demonstravimą, programos teksto skaitymą ir veikimo sekimą), arba tuos, kuriuos taikant studentams buvo sudarytos sąlygos užsiimti aktyvia veikla (darbą grupėse bendradarbiaujant, durstinį, diskusijas, minčių lietu, programavimą porose). O neigiamai įprastai vertintos paskaitos ir vizualizavimo metodai (sąvokų žemėlapis, algoritmų ir programų vizualizavimas).



24 pav. Studentų požiūris į taikytus mokymosi metodus

Analizuojant studentų požiūrį į taikytus mokymosi metodus pagal mokymosi stilių grupes (žr. 25 paveikslą) nustatyta, kad veiklos tyrimo metu, ugdant pradedančiųjų programavimo gebėjimus, taikyti mokymosi metodai labiausiai atitiko teoretikų poreikius, o mažiausiai – mąstytojų ir neišreikšto mokymosi stiliaus (t. y. nė vieno stipriai ar labai stipriai išreikšto mokymosi stiliaus tipo) grupės. Teoretikų teigiami vertinimai sudarė daugumą (80,9 proc.), o neigiami – tik dešimtadalis (10,6 proc.). Mąstytojai ir teigiamų (34,7 proc.), ir neigiamų (29,8 proc.) vertinimų skyrė apie trečdalį. Kiek daugiau kaip trečdalį neigiamų vertinimų (34,5 proc.) skyrė respondentai, kurių mokymosi stilius neišreikštas. Aktyvistų, pragmatikų ir keletą mokymosi stilių

turinčiųjų grupėse vertinimai pasiskirstė panašiai: teigiamai – kiek daugiau kaip 40 proc., neigiamai – tarp 15 ir 21 proc.



25 pav. Studentų požiūris pagal mokymosi stilių tipus

Atliekant veiklos tyrimo metu taikytų mokymosi metodų vertinimų analizę nustatyta, kiek vertinimų prieštarauja sudarytam MSMMD modeliui. Prieštaraujančiais laikomi tokie vertinimai, kai: 1) atitinkamos mokymosi stiliaus grupės respondentas metodą vertino teigiamai, nors modelyje metodas jam nurodytas kaip netinkamas, 2) atitinkamos mokymosi stiliaus grupės respondentas metodą vertino neigiamai, nors modelyje metodas jam nurodytas kaip tinkamas. Toliau analizuoti mokymosi metodų vertinimai pagal mokymosi stilių grupes ir įvertinti prieštaravimai MSMMD modeliui (žr. 26 lentelę).

Veiklos tyrime dalyvavo du studentai, turintys labai stipriai, ir vienas studentas, turintis stipriai išreikštą aktyvisto mokymosi stilių. Aktyvistai pateikė 69 mokymosi metodų vertinimus, iš kurių 19 prieštaravo MSMMD modeliui. Dažniausiai neigiamai aktyvistai vertino paskaitas, o teigiamai – demonstravimą. Pastebėta, kad MSMMD modelyje numatytų tam tikrų mokymosi metodų (diskusijos, minčių lietaus, pristatymų, programavimo porose, paskaitos, sąvokų žemėlapis, vaizdo pamokos) vertinimai atitiko modelį. Darbas grupėse bendradarbiaujant, durstinys ir varžybos surinko po vieną neigiamą prieštaraujantį modeliui įvertinimą. Teigiamų vertinimų surinko algoritmų ir programų vizualizavimas, demonstravimas, individualus darbas ir savikontrolės testai, kurie modelyje nėra priskirti aktyvisto mokymosi stiliui.

Mokymosi metodų vertinimai* pagal mokymosi stilius MSMMD modelyje

Kodas	Mokymosi metodai	Vertinimai															Bendras vertinimų kiekis	Prieštaraujama MSMMD modeliui			
		Aktyvistai			Mąstytojai			Teoretikai			Pragmatikai			Bendras vertinimų kiekis	Prieštaraujama MSMMD modeliui						
		1	2	3	1	2	3	1	2	3	1	2	3								
MM01	Algoritmų ir programų vizualizavimas	1	0	4	4	3	3	x			1	1	2	x			1	3	0	23	9
MM04	Darbas grupėse bendradarbiaujant	1	3	3	3	6	8	x			1	0	4				1	0	1	31	9
MM05	Problemų sprendimas	0	0	1	0	1	0	x			0	0	0	x			1	0	1	4	1
MM06	Demonstravimas	0	1	5	0	5	3	x			0	0	4	x			0	1	3	22	5
MM07	Diskusija	0	1	3	2	2	3	x			0	1	3				0	0	2	17	5
MM08	Programavimas porose	0	4	2	3	4	4	x			0	1	4				0	0	2	24	9
MM09	Durstinys	1	0	0	0	1	0	x			0	0	0				0	0	2	4	3
MM10	Programos teksto skaitymas ir veikimo sekimas	2	4	1	0	1	10				0	0	4				0	0	4	26	15
MM11	Individualus darbas	2	2	3	1	6	4				0	0	4	x			2	2	2	28	9
MM12	Minčių lietus	0	1	1	0	2	0	x			0	0	0				0	2	2	8	2
MM13	Paskaita	4	5	0	10	5	0				0	1	5	x			0	6	0	36	0
MM15	Pristatymai	0	3	2	0	5	5	x			0	0	4				0	4	2	25	11
MM16	Savikontrolės testas	0	2	3	5	1	3				0	0	4				0	1	1	20	10
MM17	Savokų žemėlapis	0	1	0	1	0	0	x			0	0	0				2	0	0	4	1
MM18	Vaizdo pamoka	1	0	0	3	1	0	x			1	0	0	x			0	0	0	6	4
MM21	Varžybos	1	0	1	3	1	0	x			2	0	0				0	0	0	8	1
	Iš viso	13	27	29	35	44	43				5	4	38				7	19	22	286	94

* Mokymosi metodų vertinimai: 1 – neigiamas (netinka, nepatinka, nelabai patinka); 2 – vidutinis (neblogai, nei patinka, nei nepatinka); 3 – teigiamas (tinka, patinka, labai patinka).

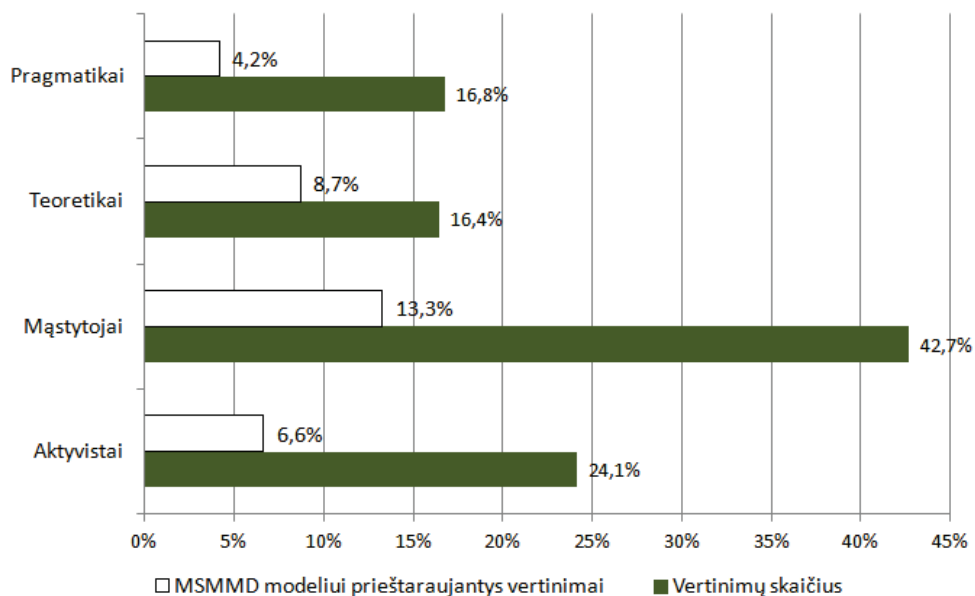
Penki veiklos tyrimo dalyviai turėjo stipriai išreikštą mąstytojo mokymosi stilių. Jie pateikė 122 mokymosi metodų vertinimus (žr. 26 lentelę). Trečdalis (38 vertinimai) mąstytojų vertinimų prieštarauja MSMMD modeliui. Dažniausiai neigiamai mąstytojai vertino paskaitas, o teigiamai – programos teksto skaitymą ir veikimo sekimą. Pastebėta, kad modelyje numatyti mokymosi metodai (algoritmų ir programų vizualizavimas ir vaizdo pamokos) surinko daugiau neigiamų vertinimų negu teigiamų. Kitų modelyje numatytų mokymosi metodų (darbo grupėse, demonstravimo, diskusijos, programavimo porose) vertinimai labiau atitinka MSMMD modelį, tačiau yra ir prieštaraujančių. Teigiamų vertinimų surinko mąstytojo mokymosi stiliui nepriskirti mokymosi metodai: individualus darbas, pristatymai, programos teksto skaitymas ir veikimo sekimas.

Du veiklos tyrimo dalyviai turėjo stipriai ir labai stipriai išreikštą teoretiko mokymosi stilių. Jie pateikė 47 mokymosi metodų vertinimus (žr. 26 lentelę), iš kurių kiek daugiau kaip pusė prieštarauja MSMMD modeliui. Dažniausiai teigiamai teoretikai vertino paskaitas. Pastebėta, kad teoretikams numatytų mokymosi metodų (demonstravimo, individualaus darbo, paskaitos) vertinimai neprieštarauja modeliui. Neigiamų vertinimų teoretikai skyrė tik 5, du iš jų – varžybų metodui ir po vieną – algoritmų ir programų vizualizavimui, darbui grupėse bendradarbiaujant ir vaizdo pamokoms. Teigiamų vertinimų surinko teoretikams nepriskirti mokymosi metodai: darbas grupėse bendradarbiaujant, diskusija, pristatymai, programavimas porose, programos teksto skaitymas ir veikimo sekimas, savikontrolės testas.

Veiklos tyrime dalyvavo du stipriai išreikštą pragmatiko mokymosi stilių turintys studentai. Jie pateikė 48 mokymosi metodų vertinimus (žr. 26 lentelę), iš jų ketvirtadalis prieštarauja MSMMD modeliui. Šioje duomenų grupėje nė vienas mokymosi metodas nesurinko daugiau kaip dviejų neigiamų įvertinimų. Daugiausia teigiamų vertinimų pragmatikai skyrė programos teksto skaitymui ir veikimo sekimui. Pragmatikams modelyje priskirtų mokymosi metodų (demonstravimo, diskusijos, programos teksto skaitymo ir veikimo sekimo, savikontrolės testo) vertinimai neprieštarauja modeliui. Individualus darbas, sąvokų žemėlapis, darbas grupėse bendradarbiaujant, algoritmų ir programų vizualizavimas ir problemų sprendimas surinko po vieną ar du neigiamus įvertinimus. Po keletą teigiamų vertinimų pragmatikai skyrė mokymosi metodams (darbui grupėse, durstiniui, minčių lietuvi, pristatymams, programavimui porose, sąvokų žemėlapiui), kurie modelyje nepriskirti pragmatiko mokymosi stiliui.

Apibendrinant nustatyta, kad 32,87 proc. sukauptų vertinimų prieštarauja MSMMD modeliui (žr. 26 paveikslą). Teoretikų grupėje pusė vertinimų (8,7 proc. iš

16,4 proc.) prieštarauja MSMMD modeliui; mąstytojų grupėje – trečdalis, o aktyvistų ir pragmatikų – tik apie ketvirtadalį.



26 pav. Prieštaraujančių MSMMD modeliui vertinimų pasiskirstymas pagal mokymosi stilių tipus

Siekiant išsiaiškinti mokymosi metodų teigiamų ar neigiamų vertinimų priežastis, buvo analizuojami veiklos tyrimo dalyvių interviu, atsižvelgiant ir į tyrėjos stebėjimų protokolų duomenis. Imant interviu, tyrimo dalyvių buvo prašoma apibūdinti taikytus mokymosi metodus, nurodant priežastis, kodėl metodas buvo naudingas ar nenaudingas. Analizės rezultatai aptariami 3.2.2 skyriuje.

3.2.2. Studentų požiūris į taikytus mokymosi metodus

Mokymosi metodo taikymo įvertinimą gali lemti įvairūs veiksniai. Mokslininkų J. Laužiko (1993), G. Butkienės, A. Kepalaitės (1996) ir kt. nuomone, mokymosi motyvacija yra pagrindinė priemonė pasiekti gerų mokslo rezultatų ir stiprinti besimokančiųjų pažangumą (Laužikas, 1993; Gage, Berliner, 1993; Barkauskaitė, Motiejūnienė, 2004). Išskiriami vidiniai ir išoriniai motyvacijos veiksniai. Vidinių veiksnių ištakos

glūdi pačioje asmenybėje, tai poreikiai, vertybės, tikslai, siekiai, interesai, emocijos. Išorinių veiksnių stimulai yra išorinės aplinkybės: studijų dalykas, dėstytojo asmenybė, mokymosi aplinka ir kt. Kiekvienas žmogaus poelgis motyvuojamas tiek vidiniais, tiek išoriniais veiksniais, mat šie veiksniai glaudžiai tarpusavyje siejasi ir vieni kitus veikia (Rupšienė, 2000). Daroma prielaida, kad studentų turima mokymosi motyvacija turėjo įtakos vertinant veiklos tyrimo metu taikytus mokymosi metodus.

Disertacinio tyrimo teorinėje dalyje analizuota N. J. Entwistle (2004) mokymosi stilių teorija susijusi su besimokančiųjų motyvacijos vertinimu, tačiau tyrimui pasirinkta mokymosi stilių tipologija motyvacijos vertinimo neapima. Todėl, renkant kokybinio tyrimo duomenis ir juos analizuojant, domėtasi ir studentų turima mokymosi motyvacija. Nustatant ją, remtasi tyrėjos (T) stebėjimų įrašais ir studentų (L1, L2, ..., L9; V1, V2, ..., V20) išsakyta nuomone apie programavimo mokymąsi. Išskirti mokymosi motyvacijos kriterijai: 0 – mokymosi motyvacija silpna, labai silpna; 1 – mokymosi motyvacija vidutinė; 2 – mokymosi motyvacija stipri, labai stipri. Išskirti pagrindžiantys teiginiai pateikti 27 lentelėje. Tiriamuoju laikotarpiu antrajame ir trečiajame veiklos tyrimo cikluose interviu davusių informantų motyvacija mokytis programuoti buvo įvairi: 4 informantų – silpna, 5 – vidutiniška ir likusiųjų 5 – stipri arba labai stipri.

27 lentelė

Mokymosi motyvacijos įvertinimas

Kategorija	Kriterijai	Teiginiai
Besimokančiųjų motyvacija	Labai silpna, silpna (0)	„Ne mano ateitis, man nepatinka“ (L3); „Na, aš papildomai nesimokiau. Tik tiek, kiek per paskaitas, nes man vienam sunku prisiversti“ (V8); „Nemanau, kad man to (programavimo – V. J.) reikia“ (V9); „Nesiklausė, uždavinių nespėdė, kitokiais reikalais užsiėmimo metu rūpinosi“ (T apie L4).
	Vidutinė (1)	„Programavimas man svarbus, bet aš programuotoju nenorėčiau būti, bet kas ten žino...“ (L2); „Programų kūrimas – labiau hobis, viena maža dalis, aš sieju savo gyvenimą su kompiuteriais, bet nebūtinai su programavimu“ (V7); „Patinka, bet nepasakyčiau, kad labai“ (V20); „Labai gerai dirbo. Kadangi visas užduotis išspėdė greitai, pasiūliau sunkesnę pamąstymui... atsakė, kad jam to nereikia, užteks to, ką padarė. Nemotyvavo net pažadėtas atlygis“ (T apie L6); „Ne visi gimsta programuotojais. Ir aš pati žinau, kad man nelabai sekasi, ir labai stebiuosi žmonėmis, kuriems tai patinka ir kuriems tikrai sekasi. Na, aš noriu gauti gerą įvertinimą, bet nelabai man patinka“ (L9).

Kategorija	Kriterijai	Teiginiai
	Labai stipri, stipri (2)	<p>„Man patinka, jeigu užkabina, galiu ir naktį atsikelti, galiu ryte pusryčiaudama galvoti apie tai, galiu atėjusi į universitetą per kažkokią paskaitą pradėti spręsti... kol neišsiaiškinu, tol nenusiraminu“ (L1);</p> <p>„<...> pastebėjau, kad iš tikrųjų iš visų dalykų, ką dabar studijuojau, gal įdomiausias dalykas yra programavimas“ (L8);</p> <p>„Programavimas man – tai kažko naujo kūrimas. Man patinka kurti. Mėgstu kažką savo sukurti, o programavimas suteikia daug erdvės kūrybai“ (V4);</p> <p>„Man tiesiog įdomu išmokyti programuoti, tai man pramoga“ (V10);</p> <p>„Man tai galimybė užsidirbti pinigus, juk labai perspektyvu, labai rinkoje mažai programuotojų, kiek aš girdžiu, nuolat viešoje erdvėje skelbiama“ (V18);</p> <p>„Labai susierzino, kad nepavyko per pratybas užbaigti darbo, bet į dienos pabaigą atsiuntė paštu užbaigtą programą. Laiške pasidžiaugė, kad pats sugalvojo“ (T apie V18).</p>

Analizuojant studentų išsakytus mokymosi metodų apibūdinimus, buvo ieškoma frazių, įvardijančių priežastis, kodėl, jų manymu, atitinkamas mokymosi metodas mokantis buvo naudingas arba nenaudingas. Toliau pateikiama kokybinių duomenų analizės apžvalga pagal veiklos tyrime taikytus mokymosi metodus.

MM01: Algoritmų ir programų vizualizavimas. Taikant metodą buvo naudojamos blokinės schemas, struktūrogramos ir algoritmo veikimo animacijos. Tyrimo metu buvo išskirti du algoritmų ir programų vizualizavimo metodo taikymo būdai: 1) kai vizualizavimą atliko dėstytojas; 2) kai vizualizavimą atliko studentai. Algoritmų ir programų vizualizavimą, kai schemas ir struktūrogramas paskaitos metu braižė dėstytojas, studentai dažniausiai vertino teigiamai: „*patinka, kai ir lentoje pabraižote, man būna aiškiau dauguma dalykų*“ (L1); „*labiau įsimintina, kadangi viskas vizualiai išdėstyta, yra žymiai geriau*“ (V8); „*efektyviau, nes gali pats įsivaizduoti, kaip viskas vyksta*“ (V10). Tačiau kai algoritmų ir programų vizualizavimo užduotis buvo skiriama studentams, metodo vertinimai buvo dažniau neigiami arba vidutiniai.

Algoritmų ir programų vizualizavimo metodo vertinimai (žr. 26 lentelę) pasiskirstė taip: neigiamų vertinimų – 30,4 proc., vidutinių – 30,4 proc., teigiamų – 39,1 proc. Daugiau kaip trečdalis šio metodo vertinimų prieštarauja MSMMMD modeliui. Analizuojant pragmatikų vertinimus nustatyta, kad jie modeliui neprieštaravo, o aktyvistų, mąstytojų ir teoretikų vertinimai modeliui prieštaravo. Algoritmų ir programų vizualizavimas MSMMMD modelyje numatytas mąstytojo ir teoretiko mokymosi stilių tipams. Analizuojant teoretikų ir mąstytojų nurodytas priežastis pastebėta, kad mąstytojų ir teoretikų išsakytos nuomonės gana prieštaringos. Nepaisant to, kad aktyvistams šis metodas nenumatytas MSMMMD modelyje, tačiau dauguma aktyvistų

vertinimų buvo teigiami. Vis dėlto, analizuojant interviu duomenis, tokio vertinimo priešasčių nepavyko nustatyti.

28 lentelė

Algoritmų ir programų vizualizavimo metodo vertinimų kategorizavimas

Kategorija	Subkategorija	Teiginiai
Besimokančiojo poreikiai	Vizualusis informacijos priėmimo būdas	„Man padėjo, nes pati programos veikimo vizualizacija padeda suprasti, kaip ten kas veikia“ (V18); „Labiau įsimintina, kadangi viskas vizualiai išdėstyta, yra žymiai geriau“ (V8); „Efektyviau, nes gali pats įsivaizduoti, kaip viskas vyksta“ (V10); „Tai irgi geras dalykas, ypač jeigu tavo vaizdinė atmintis. Pas mane kaip tik vaizdinė“ (L9); „Šitas būdas labai geras, nes pas mane vaizdinė atmintis, tai man vienas iš priimtinausių būdų“ (V7).
	Nevizualusis informacijos priėmimo būdas, neišlavintas gebėjimas vizualizuoti informaciją	„Nesuprantu aš tų schemų, nebent tik kokius ciklus aiškintis galima būtų“ (L8); „Na, aš nuo vaikystės supratau vieną dalyką, kad tai, ką aš įsivaizduoju, tikrai niekada man nepavyks to nupiešti. Na, būna kartais veikimo schemą užsirašau žodžiais, pavyzdžiui, pirmas etapas toks, antras toks..., tai man padeda sugalvoti kažką, bet aš schemas laikau galvoje, man taip paprasčiau“ (V4).

Artariant su informantais vizualizavimo užduotis išryškėjo vaizdinės informacijos priėmimo skirtumai (žr. 28 lentelę). Vieni informantai teigė, kad: „*Tai irgi geras dalykas, ypač jeigu tavo vaizdinė atmintis*“ (L9); „*Šitas būdas labai geras, nes pas mane vaizdinė atmintis, tai man vienas iš priimtinausių būdų*“ (V7). Kitiems vizualizavimas nebuvo būtinas: „*Nesuprantu aš tų schemų, nebent tik kokius ciklus aiškintis galima būtų*“ (L8); „*Na, aš nuo vaikystės supratau vieną dalyką, kad tai, ką aš įsivaizduoju, tikrai niekada man nepavyks to nupiešti. Na, būna kartais veikimo schemą užsirašau žodžiais, pavyzdžiui, pirmas etapas toks, antras toks..., tai man padeda sugalvoti kažką, bet aš schemas laikau galvoje, man taip paprasčiau*“ (V4). Interviu duomenys rodo, kad studentai vaizdinę informaciją priima skirtingai. Galima būtų teigti, kad skiriasi studentų mokymosi stiliai, tačiau tyrime naudojama mokymosi stilių tipologija neišryškina skirtingų informacijos priėmimo būdų. Gauti šio metodo apibūdinimai pagal mokymosi stilius gana prieštaringi. Vadinasi, tyrimui pasirinkta mokymosi stilių tipologija neleidžia teisingai įvertinti algoritmų ir programų vizualizavimo metodo tinkamumo mokymosi stilių tipams.

MM04: Darbas grupėse bendradarbiaujant. Darbo grupėse metodus naudinga taikyti ugdantis ne tik dalykines, bet ir bendrąsias – socialines, vadybines, mokėjimo mokytis – kompetencijas. Mokantis bendradarbiaujant ugdomi įvairūs gebėjimai: bendravimo, užduočių analizės, sprendimų priėmimo, realizuotų sprendimų pristatymo, kritinio mąstymo, refleksijos, savo ir grupės darbo įsivertinimo. Atliekant tyrimą buvo taikomi keli skirtingi darbo grupėse metodai: durstinys, diskusijos, programavimas porose. Kai kuriais atvejais darbas grupėse buvo derinamas su individualiomis veiklomis (tam tikros veiklos atliekamos grupėse, kitos – individualiai). Atskirai vertinta, kaip taikytas programavimas porose ir diskusijų metodas. Durstinio metodas taikytas tik keletą kartų, todėl jo vertinimai analizuojami kartu su kitais darbo grupėse bendradarbiaujant metodų vertinimų atvejais.

Darbas grupėse bendradarbiaujant paprastai vertintas (žr. 26 lentelę) teigiamai – 51,6 proc., o neigiamai – 19,4 proc. Modeliui prieštaraujantys vertinimai sudaro 29 proc., metodas tinkamas aktyvistams ir mąstytojams, atitinkamai jie teigiamai vertino tris kartus dažniau nei neigiamai. Tik vienas informantas, turintis stipriai išreikštą aktyvisto mokymosi stilių, apibūdindamas metodą, nurodė, kad metodas atitiko jo poreikį bendrauti: „Galėjau laisvai savo mintis reikšti, gal jos ne visuomet buvo teisingos“ (L1). Tačiau jis taip pat pabrėžė silpną grupės narių mokymosi motyvaciją: „Dirbant jautėsi ta apatiška jų būseną, joms (grupės narėms – V. J.) galbūt nepatinka tas programavimas, aš nežinau, kaip yra iš tiesų, nes sakydavo: tai gal tu padaryk“ (L1). Analizuojant neigiamus vertinimus pastebėta, kad juos galėjo nulemti nesėkmingo darbo grupėse patirtis. Nesėkmės dažniausiai buvo susijusios su nevienodu grupės narių įsitraukimu į darbą: „Buvo tokia grupė, kai pasilikau viena ir turėjau dirbti viena, kiti tik plepėjo...“ (L9) arba grupės narių negebėjimu bendrauti: „Buvo kolega, labai nebendruojantis, netgi bandžiau jo paklausti, ar tu supranti, ar nesupranti, o jis: suprantu. Ta prasme, žmogus visiškai nelinkęs į bendravimą, tai kaip ir nebuvo to grupinio darbo“ (V20). Nebendradarbiaujantys ir nebendruojantys grupės nariai – labai komplikuojantis veiksnys dirbant grupėmis, tai pabrėžė dauguma informantų. Nors teoretikams ir pragmatikams priimtinesnis individualus darbas, tačiau jie, vertindami darbą grupėse, jį vertino daugiau teigiamai. Analizuojant jų pasisakymus nustatyta, kad teigiamų vertinimų priežastys susijusios su sėkmingu darbu grupėse. Visų teigiamus vertinimus skyrusių teoretikų ir pragmatikų programavimo gebėjimai silpni ir vidutiniai, todėl suprantama, kad toks darbo pobūdis jiems paprastai būdavo naudingas. Apibūdindami teigiamą darbo grupėse patirtį jie minėjo dalijimosi patirtimi naudą: „Grupėse patiko, nes lengviau, daugiau stengiesi ir pats, o jeigu nesupranti, kitas gali paaiškinti“ (L6). Informantų nurodyti metodo vertinimo veiksniai pateikti 29 lentelėje.

Darbo grupėse bendradarbiaujant vertinimų kategorizavimas

Kategorija	Subkategorija	Teiginiai
Mokymosi metodo galimybės	Sėkmingas grupės formavimas	„Efektyviausia... taip dirbti patiko, aš kažko nežinojau, V5 žinojo, po to V20 dar pabaigė. Gera buvo komanda“ (V10); „Tąkart visi pasiūlėme, ir aš vienąkart pasiūliau, kaip daryti, V10 kaip tik „ifą“ buvo vakare pasimokinęs, todėl padėjo „ifą“ pritaikyti ir „išvartėm“ kartu (kartu atlikome užduotį – V. J.)“ (V9); „<...> su ja jaučiausi lygiavertė“ (L2).
	Galimybė mokytis iš kitų, dalijimasis patirtimi	„Grupėse patiko, nes lengviau, daugiau stengiesi ir pats, o jeigu nesupranti, kitas gali paaiškinti“ (L6); „<...> kai yra grupėje nors vienas suprantantis, tai gali ir kitiems paaiškinti“ (L3); „Kai keliese analizuoji tą programą, tai kaip ir daugiau išgirsti tų variantų, daugiau uždavinio sprendimo būdų, vienas iš kito gali pasimokyti“ (V7); „<...> kartais pritrūksta mini idėjos, kaip ten turėtų būti, ir vat kažkas turi tą <...> ir tada aš jau sakau: aaa žinau, kaip čia reikia spręsti“ (V20); „<...> tada L8, taip, ji buvo lyderė, ir ji labai gerai ir man, ir kitiems mokėjo pasakyti, kur mes klydome, ir tada, kai ji sugalvodavo sprendimą, tai ji ir mums paaiškino labai elementariai. Ir mes galėjom iš jos kažko išmokyti“ (L1).
	Aktyvus besimokančiojo vaidmuo	„Aš vis siūliau siūliau kažką, ir tai panaudojom, ir tai... dar pasiūliau kažką apskaičiuoti, tada mane kažkaip sudomino ir motyvavo kažką veikti“ (L4).
Besimokančiojo poreikiai	Silpna motyvacija	„Mane nelabai kažkaip motyvavo darbas <...>“ (L4).
	Silpni programavimo gebėjimai	„<...> o aš galbūt blogiausiai iš jų visų moku“ (L4).
	Bendradarbiavimo kompetencijos trūkumas	„Dėl to darbo su grupėmis <...>, pavyzdžiui, buvo tokia grupė, kai pasilikau viena ir turėjau dirbti viena, kiti tik plepėjo, o aš turėjau dirbti viena“ (L9); „Nepatinka darbas grupėse, man labiau patinka dirbti individualiai, nes dirbant grupėse būna, kad kiti nemąsto nieko, o tik iš manęs kažko tikisi“ (L8); „Kai daug žmonių, tada ne visi įsijungia“ (L2); „Nelabai kas ir gavosi, nebuvo grupės lyderio“ (V8); „Dirbant jautėsi ta apatiška jų būseną, joms galbūt nepatinka tas programavimas, aš nežinau, kaip yra iš tiesų, nes sakydavo: tai gal tu padaryk“ (L1).
	Bendravimo kompetencijos trūkumas, nenoras ar negebėjimas bendrauti	„Prisimenu, bet čia buvo dar tik pati pradžia mokslų, dar niekas pernelyg nebendravo“ (V8); „Buvo kolega, tada labai nebendraviantis, netgi bandžiau jo paklausti, ar Tu supranti, ar nesupranti, o jis: suprantu. Ta prasme, žmogus visiškai nelinkęs į bendravimą, tai kaip ir nebuvo to grupinio darbo“ (V20).
Mokymosi metodo ir mokymosi stiliaus dermė	„Ji man tiesiog aiškina, diskutuojam, <...>, patiko, kad galėjau pasakyti, ką galvoju“ (L2); „Galėjau laisvai savo mintis reikšti, gal jos ne visuomet buvo teisingos“ (L1).	

Organizuojant darbą grupėse, didžiausias iššūkis tyrėjai buvo grupės formavimo užduotis. Galima teigti, kad suformuoti darbo grupes, kurioms labai pasisekė (t. y. grupė pasiekė tai, ką užsibrėžė), pavyko tik keletą kartų. Nepaisant to, tyrėjos nuomone, bendradarbiaujančių grupių metodas ugdantis programavimo gebėjimus yra gera mokymosi forma tam tikrose situacijose. Ypač sėkmingu galima laikyti grupėse išbandytą dëlionės metodą, kuris taikytas aiškinant programos skaidymą į dalis. Pastebėta, kad jei tik padaugėja abejonių dėl užduoties atlikimo ir atsiranda poreikis dalytis darbą, darbo sėkmę visuomet lemia gerai suformuota grupė. Sakykime, skiriama griežtai apibrėžta užduotis, ir grupė jau turi patirties atlikti tokias užduotis. Tuomet galima tikėtis, kad grupei pasiseks, kad ir kaip ji būtų suformuota. Tačiau jei grupei pateikta užduotis dviprasmiška ir grupė neturi patirties, kuria galėtų pasiremti, pasisekti gali tik tuomet, jei grupė gerai suformuota ir dirba bendradarbiaudama. Apibendrinant darbo grupėse patirtį galima teigti, kad studentams labiausiai trūko vadybinės kompetencijos ir kitų darbo grupėse įgūdžių: bendravimo, bendradarbiavimo, kritinio vertinimo ir kt. Kai kurie veiklos tyrimo dalyviai vietoje darbo grupėse rinkdavosi individualų darbo pobūdį ir net nemėgindavo prisidėti prie bendro grupės darbo. Viena vertus, nenoras dirbti grupėje gali būti susijęs su studento mokymosi stiliumi, kita vertus, – su programavimo gebėjimais. Jau mokantieji programuoti paprastai nori savarankiškai atlikti užduotį. Tie, kurių programavimo gebėjimai silpni, atvirkščiai, nori dirbti grupėje ir turėti kolegą, galintį kompetentingai patarti: „*Grupėse dirbant svarbu, kad į grupę papultų bent vienas žmogus, kuris nors kiek „susigaudo“ <...>, duoty silpnesniam pasireikšti, o pats pakoregavimais davestų iki tam tikro tikslo*“ (V18). Kai kuriais atvejais tie, kurių gerų programavimo gebėjimai geri, prisiimdavo grupės lyderio vaidmenį ir sėkmingai vadovaudavo bendram darbui.

MM08: Programavimas porose. Tai atskiras darbo grupėse atvejis, kai grupė sudaroma iš dviejų narių. Sprendžiama programavimo užduotis atliekama dviese prie vieno kompiuterio. Abu dalyviai diskutuodami kuria ir testuoja tą pačią programą. Vienas iš jų yra vedėjas – jis sėdi prie klaviatūros ir renka programos tekstą; kitas yra tyrinėtojas – jis stebi vedėjo darbą, stengiasi pastebėti potencialias klaidas, pataria, siūlo alternatyvas. Tikslinga programos teksto rinkimą patikėti silpnesnių programavimo kalbos sintaksės vatojimo įgūdžių turinčiam studentui. Tyrinėtojas šiuo atveju atlieka mokytojo vaidmenį, todėl turi gebėti aiškinti ir, kai reikia, patarti savo porininkui. Tad renkant poras būtina atsižvelgti į studentų programavimo gebėjimus. Taikant šį metodą užduotį galima atlikti greičiau, padaryti mažiau klaidų, formuoti teigiamą nuostatą bendradarbiavimo atžvilgiu.

Programavimas porose dažniausiai vertintas (žr. 26 lentelę) teigiamai (50 proc.). Tik šiek tiek daugiau kaip dešimtadalis vertinimų buvo neigiami (12,5 proc.). MSMMD modelyje programavimo porose metodas priskirtas aktyvisto ir mąstytojo mokymosi stiliams, tačiau teigiamai įvertintas visose mokymosi stilių grupėse. Aktyvistų vertinimai modeliui neprieštaravo, o mąstytojai pateikė ir teigiamų, ir neigiamų vertinimų. Tad, išanalizavus pateiktus apibūdinimus, mokymosi metodo ir mokymosi stilių dermės veiksnys neišaiškintas. Informantai apibūdindami metodą jį nuolat lygino su kitais darbo grupėse atvejais, ir labai dažnai akcentavo, kad darbas porose sėkmingesnis negu didesnėse grupėse. Programavimo porose teigiamų ir neigiamų vertinimų veiksniai pateikti 30 lentelėje.

30 lentelė

Programavimo poromis metodo vertinimų kategorizavimas

Kategorija	Subkategorija	Teiginiai
Mokymosi metodo gali-mybės	Tinkamai parinkta bendradarbiaujanti pora	„Bet dirbant su L8 tai man labai padėjo, nes, visų pirma, mes abi buvom su vienu tikslu, o ji dar geriau „gaudosi“, patirties turi daugiau...“ (L9); „Būna, pavyzdžiui, aš garsiai sakau, ką galvoju, o jis (porininkas – V. J.) op ir pastebi kažką tokio, ko aš nepastebėjau, tada jis man pasako“ (L4).
	Bendramokslių mokymas	„<...> kai kuriuos dalykus ji man taip paaiškino, kad tik su ja aš tai ir supratau“ (L9); „Su V19 dirbau, tai gerai, nes jisai nesakydavo, kaip daryti, aš darydavau pats, o jisai pasakydavo, kas blogai“ (V18).
	Nesėkmingai parinkta pora	„Kartais man trūko porininko mąstymo proceso“ (L1); „Su V8 ir V2 tai buvo nesėkmingas atvejis, taip nieko gero ir ne-nuveikėm“ (V18).
	Skatinama atsakomybė už mokymosi veiklą	„Daug geriau nei grupėje, nes kai (dirba – V. J.) du žmonės, vis tiek abu turi galvoti“ (L1); „Bet dirbant su L8 tai man labai padėjo, nes, visų pirma, mes abi buvom su vienu tikslu...“ (L9).
Besimokančiojo poreikiai	Nenoras bendradarbiauti	„Na, ir dviese dirbant aš vis tiek savo mintyse viską turiu „susukti“ ir nelabai klausausi, ką kiti man sako“ (V4); „Nepatinka darbas grupėse, man labiau patinka dirbti individualiai. <...> aš nemoku paaiškinti, ką sugalvojau“ (L8).

Apibendrinant išsakytus apibūdinimus pastebėta, kad teigiamus vertinimus lėmė tinkamai parinkta pora ir darbo poroje sėkmė, o neigiamus vertinimus – nenoras bendradarbiauti, tai kai kuriais atvejais sukėlė labai skirtingi porininkų programavimo gebėjimai.

MM07: Diskusijos. Diskusijų metodas buvo taikomas paskaitos tematikai aptarti, užduotims analizuoti, studentams kilusiems neaiškumams po paskaitos išsiaiškinti arba darbo grupėse rezultatams aptarti. Analizuojant užduotis, identifikuojant problemas ir aptariant galimus sprendimus, ugdomi analitiniai ir loginio mąstymo gebėjimai, o taikant diskusijų po veiklos metodą skatinama reflektuoti ir įsivertinti atliktą darbą.

Diskusijos dažniausiai vertinamos (žr. 26 lentelę) teigiamai (64,7 proc.). MSMMD modelyje diskusijų metodas priskirtas aktyvisto ir mąstytojo mokymosi stiliams, tačiau ir šis metodas teigiamai įvertintas visose mokymosi stilių grupėse. Visi informantų nurodyti diskusijų metodo teigiamų ir neigiamų vertinimų veiksniai pateikti 31 lentelėje.

Aktyvistų vertinimai modeliui neprieštarauja. Apibūdindami teigiamą diskusijų patirtį informantai-aktyvistai nurodė, kad jiems patinka aktyvumas, dėmesys ir galimybė išgirsti daugiau idėjų. Taikant diskusijų metodą sudaromos sąlygos atsi-
skleisti būtent šioms aktyvistų savybėms: aktyviai veikti, būti dėmesio centre; tai įrodo mokymosi stiliaus ir mokymosi metodo dermę. Teoretikai, apibūdindami metodo taikymo patirtį, nurodė, kad diskutuojant išsiaiškinami neaiškumai ir kad jiems patinka išklausti kolegas. Vis dėlto, kuriant MSMMD modelį, diskusijų metodas teoretikams nebuvo priskirtas, remiantis teorijoje išsakytu apibūdinimu: „*Teoretikai mieliau dirba su abstrakčiomis idėjomis, individualiai*“ (Honey, Mumford, 2000). Tad P. Honey ir A. Mumfordo mokymosi veiklų klasifikacijoje (žr. 8 lentelę) diskusija teoretikams nėra priskirta, tačiau analizuojant teoretikų pateiktus apibūdinimus ir remiantis stebėjimų užrašais išryškėjo teigiamai vertinama diskusijų metodo galimybė – pasyviai dalyvauti diskusijose klausantis. Tai net ir neprieštarauja mokslininkų pateiktam teoretiko apibūdinimui: „*Pirmenybę teikia refleksyviai stebėjimui*“ (Honey, Mumford, 2000). Taigi galima teigti, kad teoretiko mokymosi stiliaus atstovams diskusija taip pat yra tinkamas mokymosi metodas. Remiantis šiais pastebėjimais nutarta patobulinti MSMMD modelį, diskusiją priskirti ir teoretikams.

Kiti diskusijų apibūdinimai nurodė teigiamų vertinimų veiksnius: reflektavimo, įsivertinimo, žinių įtvirtinimo ir dalijimosi patirtimi galimybes. Vis dėlto pragmatikų ir mąstytojų apibūdinimuose mokymosi stiliaus ir mokymosi metodo dermė neišryškinta.

Diskusijų metodo vertinimų kategorizavimas

Kategorija	Subkategorija	Teiginiai
Mokymosi metodo galimybės	Skatinama mąstyti, gilinamas diskutuojamos problemos supratimas	„<...> kai pats bandai paaiškinti savais žodžiais, tada labiau supranti. Skatina mąstyti, kai reikia, apibūdinti savais žodžiais“ (L9); „Aš dažniausiai net nesuprantu, ką reikia daryti, tai va būtent tuo momentu man ir reikalinga diskusija“ (L4); „<...> kuo daugiau idėjų, tuo geriau“ (L1).
	Įsivertinimo galimybė – išsiaiškinašamos žinių spragos	„<...> bediskutuodamas išsiaiškini, ko nesupratai, protingesni kolegų pasako ką nors naudingo“ (V20).
	Žinių įtvirtinimas, pakartojimas, mokymasis iš kitų	„<...> tiesiog žinių užtvirtinimas yra“ (V20); „Duoda naudos išgirsti daugiau nuomonių, prisimeni, ką pamiršai“ (L2); „Vat per paskaitą tai diskusija ir padeda išsiaiškinti“ (V18); „Irgi yra gerai, nes iš diskusijų daugiau viską prisimeni, jeigu žinai diskusijos tikslą, geriau prisimeni“ (V8).
	Nėra praktinio pabandymo galimybių	„<...> išbandyti reikia <...>, daug geriau, kai pati pamatau, kas užrašyta“ (L8). „<...> aš kai pats darau, man suprantamiau“ (V10).
Mokymosi aplinkos poreikiai	Neigiamas bendramokslių požiūris į mokymąsi	„Jeigu visa grupė į tai žiūrėtų rimčiau, tai būtų naudinga, ar mes tarpusavyje diskutuotume, ar su dėstytoja, manau, tai labai geras variantas“ (L1).
Besimokančiojo poreikiai	Negarsinis informacijos priėmimo būdas	„Gerai, bet ne visada gi supranti, kai kažkas pasakoma <...>“ (L8); „Efektinga, manau, tiems, kurie turi minčių ir nori jas išreikšti... man ne“ (V10).
	Silpna mokymosi motyvacija	„Nei šilta, nei šalta. Pati nelabai dalyvauju“ (L3).
	Mokymosi metodo ir mokymosi stiliaus dermė	„Patinka, kad kartu apsvarstom kartais ir nelabai apgalvotus pasiūlymus<...>“ (L1); „Diskutuojant galima apsvarstyti daug ir įvairių idėjų <...> tik man atrodo, kad aš daugiau kalbu, negu klausausi“ (L2); „<...> bediskutuodamas išsiaiškini, ko nesupratai, protingesni kolegų pasako ką nors naudingo“ (V20).

Keletas informantų apibūdindami neigiamus metodo vertinimus nurodė, kad garsinis informacijos priėmimo būdas jiems netinkamas. Kai kurios mokymosi stilių tipologijos pabrėžia skirtingus informacijos priėmimo būdus, tačiau tyrime naudojama tipologija jų neišryškina.

Apibendrinant galima teigti, kad aktyvistų ir teoretikų nurodyti teigiamų vertinimų veiksniai implikuoja mokymosi metodo ir mokymosi stilių dermės veiksmingumą.

MM13: Paskaitos. Atliekant veiklos tyrimą buvo skaitomos tradicinės ir interaktyviosios paskaitos. Šiame skyriuje aptariamas tradicinių paskaitų vertinimas. Jis dažniau vidutinis (46 proc.) arba neigiamas (39,6 proc.), teigiami vertinimai sudarė 13,9 proc. MSMMMD modeliui prieštaraujančių vertinimų nebuvo. Remiantis sudarytu MSMMMD modeliu yra žinoma, kad paskaitos metodas yra tinkamas tik teoretiko mokymosi stilių turintiems studentams. Tai parodė ir tyrimas – visi teigiami metodo vertinimai skirti studentų, kurie turi stipriai išreikštą teoretiko mokymosi stilių. Teoretikų teiginiai („*Paskaitos patinka, sunku būtų, jeigu jų nebūtų, labai trūksta tos teorinės medžiagos*“ (L6); „*Padedą mokyti..., man patinka klausyti paskaitų*“ (V20)) leidžia manyti, kad mokymosi metodas atitinka jų mokymosi stilių, tad iš tikrųjų galima teigti, kad teoretikų mokymosi stiliaus ir mokymosi metodo dermė atsiskleidė. Kitų mokymosi stilių atstovai metodą vertino neigiamai arba vidutiniškai.

Paskaitos metodas dažnai taikomas aukštosiose mokyklose pradedančiųjų programavimo gebėjimams ugdyti. Tačiau, remiantis dėstytojų ekspertų išsakyta nuomone ir asmenine dėstyto patirtimi, galima teigti, kad tai vis dėlto netinkama strategija pradedančiųjų programavimo gebėjimams ugdyti. Remiantis disertaciniame tyrime naudota P. Honey ir A. Mumfordo mokymosi stilių teorija, paskaita – tinkamas metodas tik teoretiko mokymosi stiliaus tipui. Iš tyrime dalyvavusiųjų stipriai išreikštą teoretiko mokymosi stilių turėjo labai nedaug studentų. Tai, kad paskaita nėra tinkamas mokymosi metodas, patvirtina ir besimokančiųjų išsakytas požiūris. Dauguma studentų teigė, kad paskaita nebuvo naudinga, išskyrus tuos atvejus, kai „*buvo daugiau pavyzdžių, ir bandoma išaiškinti savais žodžiais, o ne sąvokomis <...>*“ (L9). Todėl rekomenduojama pradedantiems programavimo gebėjimus ugdyti kompiuterių klasėje, o paskaitos metodą, taikomą pristatant naujas sąvokas, reikėtų derinti su kitais aktyvų besimokančiųjų darbą skatinančiais metodais.

Analizuojant informantų išsakytus paskaitos apibūdinimus, nustatyti trijų kategorijų veiksniai: 1) susiję su mokymosi metodo teikiamomis galimybėmis; 2) studijų dalyko ypatumai; 3) susiję su besimokančiojo poreikiais ir asmeninėmis savybėmis (žr. 32 lentelę).

Tradicinių paskaitų metodo vertinimų kategorizavimas

Kategorija	Subkategorija	Teiginiai
Mokymosi metodo galimybės	Nėra praktinio išbandymo galimybės	„Sausą“ teoriją galima pasiskaityti skaidrėse, o pats svarbiausias dalykas buvo tai praktiškai pabandyti, tada ir visa ta teorija iš karto susideda galvoje...“ (V18); „<...> kol nematau viso teksto programos, kol jo neišbandau ir neanalizuoju, o man aiškina teoriją, man niekas „neįlenda“ (V10).
	Per didelis gaunamos naujos informacijos kiekis	„Man sunkiausia išsėdėti paskaitoje <...>“ (V4); „Iškart tiek daug visko, kad... nematau prasmės net gilintis“ (V8).
Studijų dalyko ypatumai	Gaunamos informacijos sudėtingumas	„Man nelabai ten buvo suprantama, gal tie, kurie labiau programavo, ir suprasdavo, ką jūs kalbėdavote <...>“ (V18).
	Daug profesinių terminų, kurie pradedančiajam nėra suprantami	„Gera, jei suprantamais žodžiais, jeigu nedaug žodžių, o daugiau pavyzdžių, ir bandoma išaiškinti savais žodžiais, o ne sąvokomis <...>, kai daug sunkių žodžių, labai greitai studentai nebeklauso“ (L9); „Jūs suprantate, jums tai lengva, bet man kokių 30 proc. suprantu, o po to susipainioju...“ (L4).
Besimokančiojo poreikiai	Dėmesio koncentracijos problemos	„Ne, nepatinka, negaliu susikoncentruoti“ (L2); „Nelabai patinka, kartais pavargstu klausyt“ (L8).
	Mokymosi metodo ir mokymosi stiliaus dermė	„Paskaitos patinka, sunku būtų, jeigu jų nebūtų, labai trūksta tos teorinės medžiagos“ (L6); „Padeda mokytis <...>, man patinka klausytis paskaitų“ (V20).
	Neatitinka turimų programavimo gebėjimų	„<...> man kaip nprogramavusiam tos paskaitos nelabai ką duodavo“ (V18).

MM06: Demonstravimas. Taikant demonstravimo metodą, sprendžiama programavimo užduotis, rodomi veiksmai analizuojami ir skaidomi į atskirus fragmentus, komentuojami. Tokiu būdu imituojamas programavimo procesas, nuosekliai pereinami visi programavimo užduoties sprendimo etapai: analizė, algoritmo sudarymas, programos teksto rašymas, derinimas ir testavimas. Kad studentai aktyviau dalyvautų pratybose, užduodami probleminiai klausimai. Neturintiems programavimo patirties toks mokymosi būdas yra labai tinkamas.

Bendras demonstravimo metodo vertinimas (žr. 26 lentelę) dažniausiai teigiamas (68,2 proc.) arba vidutinis (31,8 proc.). Neigiamų vertinimų nebuvo.

MSMMD modelyje metodas priskirtas mąstytojams, teoretikams ir pragmatikams. Pastebėta, kad mąstytojai, apibūdindami demonstravimo metodą, akcentavo nuoseklumą ir dėstytojo skiriamą dėmesį detalėms. Tai natūralu, mat kruopštumas – būdingas mąstytojų elgesio bruožas. Todėl galima teigti, kad teigiamus mąstytojų vertinimus lėmė mokymosi metodo ir mokymosi stiliaus dermė. Dauguma aktyvistų

vertinimų taip pat buvo teigiami, nepaisant to, kad šis metodas modelyje jiems nenumatytas kaip tinkamas. Aktyvistai, apibūdindami, kaip taikyti demonstravimo metodą, vis dėlto nenurodė aiškių teigiamų vertinimų priežasčių.

Remiantis tyrėjos stebėjimų užrašais, pastebėta, kad paprastai metodą teigiamai vertino studentai, kurių programavimo gebėjimai silpnesni ir vidutiniai, o tie, kurių programavimo gebėjimai geri, demonstravimą teigiamai vertino tais atvejais, kai buvo demonstruojamas sudėtingesnių problemų sprendimas, t. y. tuomet, kai sprendžiama užduotis atitiko jų turimus gebėjimus. Apibendrinti metodo vertinimų veiksniai pateikti 33 lentelėje.

33 lentelė

Demonstravimo metodo vertinimų kategorizavimas

Kategorija	Subkategorija	Teiginiai
Mokymosi metodo galimybės	Dėstymo aiškumas ir nuoseklumas	„Patinka, kai ir lentoje pabraižote, man būna aiškiau dauguma dalykų, suprantu, kaip ta programa veikti turėtų“ (L1).
Besimokančiojo poreikiai	Netinkamas mokymosi tempas	„Labai nekantravo ir rodė nepasitenkinimą, nuolat nutraukdavo bendrą darbą su užduotimi nesusijusiais komentarais...“ (T apie P7); „Demonstruojama galėtų būti lėčiau, truputį per greitai“ (L4).
	Netinkamas metodas – demonstruojama užduotis per lengva	„Demonstruojant užduoties sprendimą nesiklausė, sėkmingai išsprendė užduotį savaip, išsakė nepasitenkinimą žodžiais „kiek čia galima tą patį per tą patį aiškintis“ (T).
	Taikomas metodas atitinka turimus programavimo gebėjimus	„Patinka, kai ir lentoje pabraižote, man būna aiškiau dauguma dalykų, suprantu, kaip ta programa veikti turėtų“ (L1); „Tai va tas ir padeda, pavyzdžiui, kokio nors uždavinio sprendimas, parodymas, paaiškinimas, kas kaip, tai tikrai padeda“ (V20).
	Mokymosi metodo ir mokymosi stiliaus dermė	„Tas būdas man turbūt geriausias, nes aš tada matau, kokia panaudota funkcija, tada dar kartą dėstytojas pažymi, ką ta funkcija daro, ir kai iškart galiu pamatyti rezultatą, kad būtent taip padaro, tada geriausiai galiu suvokti“ (L9); „Taip geriau, patogiau... ir suprantamiau; blogai, jeigu kas nors praleidžiama, o čia geriau – visos smulkmenos išsiaiškinamos ir užrašomas kodas“ (V4); „Patinka, kad viskas iš anksto apsvairstoma, o po to rašoma programa“ (V10).

Apibendrinant galima teigti, kad mokymosi stilius neturi didelės reikšmės, kai ugdomas pradedančiųjų programavimo gebėjimus taikomas demonstravimo metodas, nes tiriamuoju laikotarpiu šis atitiko bene visų besimokančiųjų poreikius. Tačiau pažymėtina, kad, stebėjimų duomenimis, metodas sėkmingai taikomas tuomet, kai demonstruojamų užduočių sudėtingumas atitinka studentų turimus programavimo gebėjimus.

MM10: Programos teksto skaitymas ir veikimo sekimas. Programos teksto skaitymas ir veikimo sekimas – tai populiarus pradedančiųjų programavimo mokymosi būdas. Taikant šį mokymosi metodą ugdomas programavimo konceptų suvokimas, lavinami programos teksto skaitymo ir programos teksto supratimo įgūdžiai, padedama surasti algoritmo logines klaidas, tačiau iš besimokančiojo reikalaujama atidumo ir kantrybės. Taikant šį metodą, buvo atliekamos kelių tipų užduotys:

- pakomentuoti kiekvieną programos teksto eilutę;
- sekti, kaip programos vykdymo metu kinta kintamųjų reikšmės;
- nustatyti, kokią užduotį atlieka parašyta programa.

Bendras programos teksto skaitymo ir veikimo sekimo metodo vertinimas (žr. 26 lentelę) dažniausiai teigiamas (73,1 proc.) arba vidutinis (19,2 proc.). Kaip teigė daugelis informantų, šis mokymosi būdas dažnai buvo naudojamas mokantis savarankiškai.

Išanalizavus duomenis nustatyti programos teksto skaitymo ir veikimo sekimo metodo vertinimų veiksniai (žr. 34 lentelę).

34 lentelė

Programos teksto skaitymo ir veikimo sekimo metodo vertinimų kategorizavimas

Kategorija	Subkategorija	Teiginiai
Mokymosi metodo gali- mybės	Padedą suvok- ti programavimo konceptus	„Šitas būdas tikrai padeda, nes jau matai, kaip tai veikia“ (V8); „Kai mes analizuojame programos tekstą, tada man daug geriau sekasi suprasti, taip, pavyzdžiai yra geriau, labai patinka taip mo- kyti“ (V7).
	Padedą identi- fikuoti klaidas	„Naudoju, labai padeda suvokti, kas parašytoje programoje veikia, o kas ne“ (L8); „Kurį laiką to nedariau, bet vieną dieną pabandžiau ir supratau, kad man taip lengviau suprasti savo programavimo klaidas“ (L9).
Besimokan- čiojo porei- kiai	Mokymosi metodo ir mokymosi sti- liaus dermė	„Padeda suprasti tą patį ciklą, kaip jis veikia, viską galima iš eilės ir detaliai išsiaiškinti“ (V10); „Jeigu bent apytikriai žinai, ką ji daro, tai pamažu kažko ir išmoksti skaitydamas, gal mažus gabalėlius keisdamas, žiūrint, kas iš to bus. Aš beveik vien taip ir mokiausi...“ (V4).
	Asmeninės savybės – kantrybės – trū- kumas	„Ir taip, ir ne, dažnai neužtenka kantrybės“ (L2); „Bet aš tai galvoje, mintyse darau, nepiešiu jokių reikšmių lentelių, kantrybės piešti neužtenka“ (V4).
	Netinkamas me- todas – neatitinka programavimo ge- bėjimų	„Aš juo mažiau naudojasi, negalėčiau vertinti nei teigiamai, nei neigiamai, man ir taip viskas aišku“ (L1); „Užduotis su komentarais... nesuprantu aš to eilučių komentavimo prasmės, nieko naujo neišmokau“ (V10).

Analizuojant vertinimus pagal mokymosi stilius nustatyta, kad daugiausia teigiamų vertinimų skyrė mąstytojai, nors MSMMD modelyje šis metodas nebuvo jiems numatytas kaip tinkamas. Iš metodo apibūdinimų pastebėta, kad mąstytojai akcentavo metodo suteikiamą galimybę išsamiai, viską iš eilės išsiaiškinti: „*Padedą suprasti tą patį ciklą, kaip jis veikia, viską galima iš eilės ir detalai išsiaiškinti*“ (V10) ir „*Jeigu bent apytikriai žinai, ką ji daro, tai pamažu kažko ir išmoksti skaitydamas, gal mažus gabalėlius keisdamas, žiūrint, kas iš to bus. Aš beveik vien taip ir mokiausi...*“ (V4). Kadangi programos teksto skaitymo ir veikimo sekimo užduotys reikalauja kruopštumo, o kruopštumas būdingas mąstytojams, akivaizdu, kad derėtų šį metodą modelyje priskirti ir mąstytojams. Programos teksto skaitymo ir veikimo sekimo metodas tinkamas ir pragmatikams, jų vertinimai modeliui neprieštaravo.

Remiantis stebėjimų duomenimis nustatyta, kad komentavimo užduotis teigiamai įvertino tyrimo dalyviai, kurių programavimo gebėjimai silpni, o vidutiniškai – kurių programavimo gebėjimai geri. Užduotis identifikuoti sprendžiamą problemą pagal pateiktą programos tekstą teigiamai vertino studentai, kurių programavimo gebėjimai labai geri.

MM11: Individualus darbas. Tai darbas, atliktas sprendžiant individualias programavimo užduotis. Tokios užduotys būtinos mokantis programuoti, tačiau tinkamos jau turintiems programavimo patirties. Pradedantiesiems vis dėlto rekomenduotina individualias programavimo užduotis skirti baigiant tam tikros temos studijas. Kita vertus, turintiems programavimo patirties galima individualias užduotis skirti iš karto. Bendraujant su informantais pastebėta, kad jie suvokia tokio mokymosi naudą, vertina jį kaip būtiną, tačiau kartu ir kaip labai sunkų mokymosi būdą: „*Reikia tokių užduočių, daugiausia laiko užtrunka sugalvoti, kaip pradėti*“ (V20); „*Aišku, taip sunkiausia, tačiau man labiausiai patinka taip dirbti*“ (L8). Nustatytas atvejis, kai teigiamą individualaus darbo vertinimą lėmė neigiama darbo grupėse patirtis: „*Nesuprantamų idėjų niekas neprimetinėja, kaip kad buvo grupėse dirbant*“ (L8). Jeigu individuali užduotis yra kontrolinė (t. y. vertinama pažymiu ir turi būti atlikta užsiėmimo metu), dalis studentų patiria stresą: „*Kai yra kontrolinis, tada, be abejo, stresuoji, skubi, nes norisi kuo daugiau padaryti. Tada būna kažkaip ne iki galo, bent jau man taip būna, „užsiblokuoja“ kažkas ir negali išspręsti, o kai ateini į namus, visiškai kitą aplinką, parašai Tu tą kodą ir jokių problemų nėra*“ (V18). Pasidomėjus, kaip studentai elgiasi, kai atsiduria aklavietėje, dažniausiai minėtą problemą

sprendžia ieškodami informacijos internete arba konsultuodamiesi su dėstytoju ir bendramoksliais: „<...> žiūriu pavyzdžius, kuriuos mes sprendėme paskaitose, imu dalis, kurios panašios iš vienos programos, iš kitos <...>, o kai jau nepavyksta nieko, tada ieškau internete“ (L4).

Individualios užduotys paprastai vertinamos (žr. 26 lentelę) teigiamai (46,4 proc.) arba vidutiniškai (35,7 proc.). Dažniausiai teigiamus vertinimus skyrė teoretikai ir mąstytojai. MSMD modelyje individualus darbas kaip tinkamas mokymosi būdas būtent ir priskiriamas teoretikams ir pragmatikams. Teoretikų vertinimai modeliui neprieštaravo. Apibūdindami individualaus darbo metodą jie teigė nedarantys neapgalvotų sprendimų: „Tiesiog stengiuosi laikytis plano, kad viską padaryčiau kuo geriau“ (L6), ir nemėgstantys jaustis neužtikrintai: „Gerai, kai yra pas ką paklausti, tiesiog pasitikrinti, kad tikrai taip“ (V20). Tobulumo siekis yra būdingas teoretikams, o individualus darbas ir leidžia atskleisti savo gebėjimus. Tad galima teigti, kad teoretikams individualaus darbo teigiamų vertinimų veiksnys – mokymosi metodo ir mokymosi stiliaus dermė. Pragmatikams šis veiksnys nėra toks aktualus – jie labiau akcentavo su besimokančiojo poreikiais susijusius veiksnys.

Analizuojant duomenis nustatyta, kad dažniausiai neigiamai šį mokymosi būdą vertino neišreikšto mokymosi stiliaus atstovai. Šie informantai paprastai turi ir silpną mokymosi motyvaciją. Tai leidžia manyti, kad šiuo atveju neigiamus vertinimus labiau nulemia būtent mokymosi motyvacijos trūkumas. Išanalizavus duomenis išskirti individualių programavimo užduočių teigiamų ir neigiamų vertinimų veiksniai (žr. 35 lentelę).

35 lentelė

Individualių programavimo užduočių vertinimų kategorizavimas

Kategorija	Subkategorija	Teiginiai
Mokymosi metodo galimybės	Aktyvus besimokančiojo vaidmuo	„Gavusi užduotį, bandau nesinervinti, jeigu matau, kad nelabai suprantu užduotį, bandau po truputį suprasti... jeigu nežinau kaip, skaitau teorinę medžiagą paskaitų, gal rasiu kažkokį „kabluką“, o kai „kabluko“ nerandu, tada einu į internetą ir ten ieškau, taip kažką naujo išsiaiškinu“ (L9).
	Užduoties sudėtingumas atitinka programavimo gebėjimus	„Jeigu suprantu ir sekasi, tada įdomu. Jeigu ne – tada erzina“ (V4); „Gerai vertinu, kai užduotis įveikiama“ (L4).

Kategorija	Subkategorija	Teiginiai
Besimokančiojo poreikiai	Programavimo kalbos sintaksės žinių trūkumas	„Pirmiausiai išanalizuoju, kaip turėtų veikti, o po to sunkiausia man dalis – kai reikia viską užrašyti, niekaip neprisimenu tų visų struktūrų, kur ir ką rašyti“ (V7).
	Užsispyrimas ir kantrybė, jei pasiseka, didėja motyvacija ir savivertė	„Labai susierzino, kad nepavyko per pratybas užbaigti darbo, bet į dienos pabaigą atsiuntė paštu užbaigtą programą. Laiške pasidžiaugė, kad pats sugalvojo“ (T).
	Dėmesio koncentracijos problemos	„Kai yra kontrolinis, tada, be abejo, stresuoji, skubi <...> bent jau man taip būna, „užsiblokuoja“ kažkas ir negali išspręsti<...>“ (V18); „Kai būna skirta individuali užduotis tik man asmeniškai ir tada aš nežinau, kaip susikonscentruoti į tą užduotį <...>, tada aš jaučiu iš savęs kaip ir toki spaudimą, nes noriu kaip greičiau, kaip geriau ir kad ta programa veiktų“ (L1).
	Mokymosi metodo ir mokymosi stiliaus dermė	„Geriau, kai yra pas ką paklausti, tiesiog pasitikrinti, kad tikrai taip“ (V20); „Tiesiog stengiuosi laikytis plano, kad viską padaryčiau kuo geriau“ (L6).

MM15: Pristatymai. Individualiai arba grupėje atliktų darbų pristatymai ir vertinimai, vadovaujant dėstytojui, studentams yra labai vertingi. Jie pristatymų metu turi prisiimti pranešėjo ir dėstytojo vaidmenis, kai kurie iš jų būtent dėl to jaučiasi patenkinti. Kai studentai motyvuotai išitraukę atlieka užduotį, jie atsakingai siekia gero rezultato, todėl atlikto darbo pristatymas sukelia teigiamus jausmus. Stebint kitų pristatymus leidžiama laisvai reikšti savo nuomonę, išsakyti savus pastebėjimus ar pasiūlymus, kaip tobulinti užduoties sprendimą, užduoti klausimų, pateikti kritinių pastabų. Kai tenka vertinti kolegų pristatymus, studentai, bendradarbiaudami tarpusavyje, natūraliai mokosi prisiimti atsakomybę už vertinimo objektyvumą ir suvokia vertinimo prasmę. Tokiu būdu ugdoma kritinio mąstymo kompetencija ir gebėjimas priimti kritiką.

Pristatymus labai teigiamai įvertino darbus pristatę studentai. Taigi pristatymų metodo vertinimai dažniausiai buvo teigiami (52 proc.), kiti pristatymus vertino vidutiniškai (žr. 26 lentelę). Analizuojant teigiamų ir neigiamų vertinimų priežastis išskirti veiksniai pateikti 36 lentelėje.

Pristatymų metodo vertinimų kategorizavimas

Kategorija	Subkategorija	Teiginiai
Mokymosi metodo galimybės	Įsivertinimo galimybė, išsiaiškinamos žinių spragos	„Kai kiti pristatinėja panašų darbą, geriau supranti, kas ir kaip yra, ką padarei gerai, kur suklydai“ (L4).
	Kalbėjimo įgūdžių tobulinimas	„<...> išmoksti pateikti savo mintis, atsakyti į klausimus, tai šiais laikais, manau, yra labai svarbu“ (V18).
	Žinių įtvirtinimas, pakartojimas	„Aiškindamas kitiems pats dar geriau įsigilini, geriau supranti“ (L1); „<...> beaiškindamas kitiems ir pats geriau supranti“ (L3); „Aiškinimas kitam, aišku, padeda, nes prisimeni, ką darei, pakartoji viską dar kartą“ (V7); „Esi kaip ir dėstytojas, ir turi paaiškinti, ką tu čia darei, pamatai, kad tai ne taip jau ir lengva“ (L9).
Besimokančiojo poreikiai	Pasitenkinimas pristatant atliktą darbą	„Kai kitas aiškina, dažniausiai nesiklausai <...>, bet kai aš aiškinu, man atrodo, kad manęs visi klauso, gal tai ir iliuzija, bet man tai labai patinka“ (V4).
	Mokymosi metodo ir mokymosi stiliaus dermė	„Be to, man tai neįprasta, bet labai įdomi veikla – vertinti kitus“ (L1); „Patiko dirbti kitaip nei įprasta“ (L2).
	Silpni programavimo gebėjimai	„Būna, kad dirbam, kažką parašom, pakeičiam vietom, patys nesuprantam, bet pavyksta. Tada, kai reikia paaiškinti, nesupranti, kaip ten kas atsiranda“ (L6); „Ką gali pristatyti, jeigu nelabai supranti...“ (V8).
	Stiprina mokymosi motyvaciją	„Aiškinimas kitiems didina motyvaciją, man taip atrodo, ne žinių prideda, o būtent motyvacijos“ (V4).

Analizuojant vertinimus pagal mokymosi stilius nustatyta, kad aktyvistų vertinimai atitiko modelį. Jie nurodė, kad pristatymai buvo naudingi ir kad mokydami kitus jie ir patys išmoko daugiau. Mokymosi mokant kitus nauda aiški ir iš kitų informantų pasisakymų. Kita teigiamų vertinimų priežastis, nurodyta aktyvistų, – naujoviškas mokymosi būdas, galimybė vertinti kitus. Aktyvistams patinka darbų kaita, o kadangi pristatymų metodas suteikia būtent tokią galimybę, jis ir buvo įvertintas teigiamai. Mokymosi metodo galimybės sutampa su aktyvistų savybėmis, tad galima teigti, kad teigiamus vertinimus paskatino mokymosi metodo ir mokymosi stiliaus dermė.

Kitų informantų (mąstytojų, teoretikų ir pragmatikų) vertinimai dažniausiai buvo teigiami, nepaisant to, kad analizuojant jų apibūdinimus mokymosi metodo ir mokymosi stilių dermė neišryškėjo. Neigiamai pristatymus vertino informantai, kurie nemėgsta kalbėti viešai ir kurių programavimo gebėjimai silpni. Remiantis diskusijose teigiamai teoretikų vertinta pasyvaus dalyvavimo veikla, galima būtų manyti, kad

ir pristatymas kaip mokymosi būdas teoretikams yra tinkamas. Vis dėlto interviu duomenys to nepatvirtino.

MM16: Savikontrolės testai. Kadangi studijoms naudojama virtualioji mokymosi aplinka, studijų modulyje buvo suprojektuoti savikontrolės testai su grįžtamuju ryšiu (t. y. trumpas paaiškinimas, kodėl pasirinktas atsakymas yra teisingas ar neteisingas). Grįžtamasis ryšys laikant testą pateikiamas automatiškai, jei neteisingai atsakoma į klausimą. Testuose buvo naudojamos kelių tipų užduotys: programavimo konceptų suvokimo, programavimo kalbos sintaksės, loginio mąstymo ir programų tekstų analizės. Savikontrolės testai padeda susisteminti turimas žinias, jas įtvirtinti ir apibendrinti. Kita vertus, kadangi jie nebuvo privalomi, jais naudojosi labai maža dalis studentų.

Savikontrolės testus studentai dažniausiai vertino teigiamai (55 proc.). Metodą neigiamai (25 proc.) vertino tik mąstytojai. Visi vertinimai pateikti 26 lentelėje.

MSMMD modelyje savikontrolės testai priskirti teoretikams ir pragmatikams, nes jie labiau už kitus mokymosi stilių atstovus vertina savarankišką individualų darbą. Teoretikų ir pragmatikų pateikti metodo vertinimai modeliui neprieštaravo. Apibūdindami savikontrolės testų vertinimus pragmatikai išryškino savikontrolės testų naudingumą, galimybę vėliau pritaikyti testavimosi metu įgytas žinias: „*Sužinai teisingus atsakymus, dažniausiai vis tiek kažką iš ten panaudoti gali*“ (V7). Galima daryti prielaidą, kad vertinant pragmatikų atžvilgiu išryškėjo savikontrolės testų teigiamų vertinimų veiksnys – mokymosi metodo ir mokymosi stiliaus dermė, nes apibūdindami metodą jie pabrėžė jo praktinę naudą. Teoretikai mokymosi stiliaus ir mokymosi metodo dermės neišryškino. Kai kurie mąstytojų vertinimai prieštaravo MSMMD modeliui. Apibūdindami savikontrolės testus, jie nurodė, kad pernelyg sudėtinga sprendžiant užduotis pritaikyti teorines žinias. Teigiamai respondentai įvertino savikontrolės testų teikiamą grįžtamąjį ryšį.

Savikontrolės testų vertinimų kategorizavimas

Kategorija	Subkategorija	Teiginiai
Besimokančiojo poreikiai	Konceptualizavimo problemos; negėbėjimas susieti teorijos ir praktikos	„Man buvo testas – tai teorija, tai aš tiesiog išmokau teoriją, bet praktikoje man, kaip sakiau, sunku susieti“ (L9); „Nepatinka, ypač teoriniai klausimai. Ką ten iš to testo išmoksi“ (L8); „Man asmeniškai, kai pradėdi jį daryti ir daryti, aš išmokstu atsakymus, bet ne kurių ir kodėl neatsakau...“ (L1)
	Galimybė peržiūrėti savo padarytas klaidas	„<...> ypač gerai, kai Tu gali pasižiūrėti į savo klaidas“ (V18); „grįžtamasis ryšys <...> man patinka, sužinojau sąvokų apibrėžimus“ (L1).
	Mokymosi metodo ir mokymosi stiliaus dermė	„Padėda mokyti, nes sužinai teisingus atsakymus, dažniausiai vis tiek kažką iš ten panaudoti gali“ (V7); „<...> sužinojau sąvokų apibrėžimus, išlaikiusi testą sužinojau, ko reikia atsiskaitymui“ (L1).

MM18: Vaizdo pamokos. Vaizdo pamokos tarp informantų nebuvo populiarios – tik dalis (8 informantai) studentų bandė savarankiškai mokyti žiūrėdami įvairius vaizdo įrašus. Dauguma iš jų (6 informantai) vaizdo pamokas net ir įvertino neigiamai. Galima daryti išvadą, kad studentų savarankiškai internete surastos ir naudotos vaizdo pamokos nebuvo suderintos su mokymo planu ir atliktomis veiklomis, o tai iš esmės ir nulėmė jų nepopuliarumą. Išskirti neigiamų vaizdo pamokų vertinimų veiksniai (žr. 38 lentelę).

MSMMD modelyje vaizdo pamokos priskirtos mąstytojams ir teoretikams, nes jie teikia pirmenybę refleksyviai stebėjimui. Nepaisant to, analizuojant vaizdo pamokų apibūdinimus, išsakytus interviu metu, nenustatyta, kad vaizdo pamokų naudojimas dera su mąstytojo ar teoretiko mokymosi stiliais. Du informantai (vienas – neišreikšto mokymosi stiliaus, kitas – turintis kelis stipriai išreikštus mokymosi stilius) teigė, kad vaizdo pamokos jiems buvo tam tikrais atvejais naudingos: „*Taip, vaizdo įrašus namie žiūrėjau, man ten buvo labai aišku. Yra gerų įrašų*“ (L4); „*Žiūrėjau angliškus filmukus, nes lietuviškų nėlai ten yra, na, ir taip mokiausi. Kažkiek padėjo...*“ (V7). Kiti studentai teigė, kad vaizdo pamokos – neveiksmingas ir sunkus mokymosi būdas. Taigi paprastai informantų nurodytos nenaudojimo priežastys susijusios su asmeninėmis besimokančiųjų savybėmis.

Vaizdo pamokų vertinimo kategorizavimas

Kategorija	Subkategorija	Teiginiai
Besimokančiojo poreikiai	Netinkamas mokymosi tempo; dėmesio koncentracijos problemos, neužtenka kantrybės	„Bandžiau naudotis, tačiau man tai nepasiteisino, nes labai daug užima laiko“ (L8); „Kaip programą rašyti, YouTube esu žiūrėjusi, bet labai daug laiko užima, nelabai patinka“ (L6); „Mane vargina stabdyti ir sukti atgal <...> man labiau patinka <i>tutorial</i> (pamokėlės – V. J.), „manualai“, video negaliu žiūrėti, neršina, aš ne toks greitas“ (V4); „Ne visada užtenka kantrybės žiūrėti vaizdo pamokas“ (L9).
Mokymosi metodo galimybės	Nėra grįžtamojo ryšio, negalima gilintis ir išsiaiškinti iškilusių klausimus	„Bandžiau, iki tam tikro lygio buvo efektyvu, kol nereikia Tau ko nors individualiai paaiškinti“ (V10).
	Netenkina vaizdo įrašų turinys ir kokybė – medžiaga nesuderinta su mokymosi planu	„Dažniausiai pusė tų įrašų jau yra pažengusiems, kad „nuo nulio“ mokintų, labai mažai yra“ (V9); „Blogiausia, kai Tau reikia vieno dalyko, o Tu būtent šito nerandi, jis integruotas į kitą užduotį; ir tada visą žiūrint jie nepabrėžia, ko Tau reikia, ir tada labai sunku susivokti, „kur čia kampas““ (L9).

MM21: Varžybos. Varžybų metodas pasiteisintų ir būtų vertingas, jeigu studentų grupė susidarytų iš lygiaverčių varžovų. Deja, veiklos tyrime dalyvavusiųjų turimi programavimo gebėjimai labai skyrėsi, todėl varžybų metodas nepasiteisino ir informantai jį vertino neigiamai.

MSMMD modelyje varžybos priskirtos aktyvisto mokymosi stiliui. Analizuojant jų apibūdinimus pastebėta, kad metodas jiems tinkamas, tačiau tik tais atvejais, kai varžybose tampama nugalėtoju: „*Skatina daryt, galvot greičiau, padaryti, nesiblaškyti, bet jei nelaimi...*“ (L2); „*Manau, kad tai ir naudinga, pradedi racionaliau mąstyti, ieškoti to greičiausio varianto, o kai dar tavęs laukia prizas, kas nenorėtų laimėti*“ (L1). Remiantis mokymosi stilių tipologija žinoma, kad aktyvistai siekia pripažinimo, o darbui juos motyvuoja pažadėtas atlygis. Todėl galima teigti, kad šiuo atveju pasireiškė mokymosi stiliaus ir mokymosi metodo dermė. Kiti informantai varžybas vertino neigiamai: „*Aš tokioje situacijoje jaučiuosi blogai, nes esu labai lėtas žmogus, man reikia daugiau laiko*“ (L9); „*Tai tikra demotyvacija, labai nemėgstu, kai man riboja laiką...*“ (V4); „*Man tai nelabai toks mokymosi būdas, nes kai aš pradėdu skubėti, su kažkuo lenktyniauti, man sunkiau mąstyti...*“ (V7).

Apibendrinant informantų pasisakymus išskirti varžybų metodo teigiamų ir neigiamų vertinimų veiksniai (žr. 39 lentelę).

Varžybų metodo vertinimų kategorizavimas

Kategorija	Subkategorija	Teiginiai
Besimokančiojo poreikiai	Mokymosi metodo ir mokymosi stiliaus dermė	„Pradžioje tai labai skubėjau, visko turbūt neapmąščiau <...>, bet manau, kad tai ir naudinga, pradedi racionaliau mąstyti, ieškoti to greičiausio varianto <...>, o kai dar tavęs laukia prizas, kas nenorėtų laimėti“ (L1); „Mane tai skatina daryt, galvot greičiau, padaryti, nesiblaškyti...“ (L2).
	Asmeninės savybės, rūpinimasis kitais	„Na jau ne, lieka daug nuskriaustų“ (L8).
Mokymosi metodo galimybės	Nelygiaverčiai varžovai, mažėja mokymosi motyvacija	„<...> niekas nebenori stengtis, laimės tas, kuris daugiau moka“ (V9); „Mūsų grupėje yra žmonių, kurie labai greit parašo, tai vat tik labiau „susistresuojii“ (V18); „<...> bet mes labai skirtingo lygio ir man konkuruoti su tais, kurie labai gerai moka... nelygios varžybos, juokinga“ (V20).
	Neužtikrinama mokymosi parama	„Tai tikra demotivacija, labai nemėgstu, kai man riboja laiką. Matot, tokį ultimatumą paskelbęs dėstytojas kaip ir užkerta visus kelius kažko jo klausti ir pabandyti išsiaiškinti, turi jau „virtti savo sultyse“ (V4); „<...> jeigu pirmąkart nepasiseka, jau po to kažin ar vėl taip pulčiau, nebebūtų to stimulo“ (V10).

Apibendrinant veiklos tyrime taikytų metodų patirtį, galima teigti, kad kokybinis tyrimas leido atskleisti veiksnius, formuojančius studentų požiūrį į taikytus mokymosi metodus. Veiksniai buvo suskirstyti į keletą kategorijų:

- veiksniai, susiję su besimokančiojo poreikiais;
- veiksniai, susiję su taikomo mokymosi metodo galimybėmis;
- veiksniai, susiję su studijų dalyko ypatumais;
- veiksniai, susiję su mokymosi aplinka.

Dažniausiai studentų nuomonę apie taikytų mokymosi metodų tinkamumą lemia veiksniai, susiję su besimokančiojo poreikiais ir mokymosi metodo teikiamomis galiomybėmis. Mokymosi stiliaus ir mokymosi metodo dermės veiksnys nustatytas retai. Tiriant studentų požiūrį apie taikytus mokymosi metodus atskleisti mokymosi stiliaus ir mokymosi metodo dermės veiksnį analizuojant vien tik studentų pasisakymus yra sudėtinga. Mokymosi stiliaus ir mokymosi metodo dermė neišryškėjo algoritmų ir programų vizualizavimo, programavimo porose ir vaizdo pamokų naudojimo atvejais. Mokymosi metodo ir mokymosi stiliaus dermės veiksnys priskirtas besimokančiojo poreikių grupei.

3.2.3. Veiklos tyrimo rezultatų apibendrinimas

Apibendrinant veiklos tyrimo metu sukaup tus mokymosi metodų vertinimus ir apibūdinimus galima teigti, kad:

- Tam tikra dalis sukaup tų vertinimų prieštarauja MSMMD modeliui. Daugiausia neatitikčių užfiksuota analizuojant teoretikų (mokymosi stiliaus tipas pagal P. Honey ir A. Mumfordo mokymosi stilių tipologiją) vertinimus – daugiau kaip pusė teoretikų pateiktų vertinimų neatitinka MSMMD modelio. Kitų mokymosi stilių atvejais prieštaraujančių vertinimų nustatyta nuo 25 iki 31 proc.
- Duomenų analizės rezultatai rodo, kad mokymosi stiliaus ir mokymosi metodo dermė išryškėja tam tikrais mokymosi metodų ir mokymosi stilių derinimo atvejais:
 - aktyvistai – darbas grupėse, diskusijos, pristatymai, varžybos;
 - teoretikai – paskaitos, diskusijos (pasyviai dalyvaujant), individualus darbas;
 - mąstytojai – demonstravimas, programos teksto skaitymas ir veikimo sekimas;
 - pragmatikai – savikontrolės testai.
- Studentų požiūriu, pradedančiųjų programavimo gebėjimams ugdytis tinkamiausi šie mokymosi metodai: programos teksto skaitymas ir veikimo sekimas, programavimas porose, demonstravimas. Individualaus darbo svarba pripažįstama, tačiau dažniausiai vertinama vidutiniškai.
- Studentų nuomone, tradicinės paskaitos nėra tinkamas mokymosi metodas ugdantis programavimo gebėjimus. Veiklos tyrimu nustatyta, kad taikant paskaitos metodą išryškėja paties metodo ir studijų dalyko inspiruojami veiksniai: gaunama informacija pernelyg sudėtinga, jos pernelyg daug ir nėra galimybės išgirstas teorines žinias išbandyti praktiškai programuojant. Kita vertus, paskaitos metodas yra tinkamas tik teoretikams, o tokį mokymosi stilių turi tik maža dalis pirmojo kurso studentų.
- Algoritmų vizualizavimo metodo ir mokymosi stilių pagal P. Honey ir A. Mumfordo mokymosi stilių tipologiją atitikimas nėra pagrįstas, nes metodą iš esmės apibūdina, kaip informacija perteikiama vizualiai, o P. Honey ir A. Mumfordo mokymosi stilių tipologijoje mokymosi stiliai neskirstomi pagal vizualios / nevizualios informacijos priėmimą.
- Pasirinkta veiklos tyrimo strategija leido atskleisti studentų požiūrį į taikytus mokymosi metodus formuojančius veiksnius. Tačiau tyrėjos dalyvavimas,

renkant tyrimo duomenis, padidino tyrimo subjektyvumą. Interviu metu sukaupiti duomenys neatskleidė tyrėjai su jos kaip dėstytojos asmenybe susijusių ir tyrimui įtakos dariusių veiksnių.

- Remiantis studentų požiūriu atskleisti pozityvūs ir / ar negatyvūs mokymosi metodų taikymo aspektai sudaro prielaidas tobulinti pradedančiųjų programavimo gebėjimų ugdymosi procesą ir MSMMMD modelį. Remiantis veiklos tyrimo rezultatais, nutarta patobulinti MSMMMD modelį: diskusijų metodą priskirti visiems mokymosi stilių tipams, o programų teksto skaitymą ir veikimo sekimą – mąstytojams ir pragmatikams.

TYRIMO REZULTATŲ APTARIMAS IR DISKUSIJA

Išanalizuoti įvairių šalių mokslininkų požiūriai į mokymosi stilių teorijų pritaikymą pedagoginėje praktikoje parodė, kad dažniau laikomasi skeptiško arba kai kuriais atvejais kategoriškai neigiamo požiūrio. Pagrindinės ir daugiausia įtakos turėjusios mokymosi stilių teorijos (Kolb, Dunn & Dunn, Felder Solomon ir kt.) sukurtos XX a. pradžioje. Tad šiuo metu sudarant tipologijas jau remiamasi viena ar keliomis kitomis, sukurtomis anksčiau. Tačiau kritikai vis dėlto lig šiol akcentuoja, kad nėra konkretaus ir aiškaus mokymosi stiliaus sąvokos paaiškinimo. Nesutariama, ar tai savybės, preferencijos (besimokančiojo teikiamas prioritetas vienam ar kitam mokymosi metodui), įpročiai, strategijos ar biologiniai bruožai (Coffield ir kt., 2004; Pashler ir kt., 2009). Be to, nėra įtikinamų duomenų, įrodančių, kad ugdymo procese taikant mokymosi stilių teorijas gerėtų mokymasis, lygiai taip pat kaip nėra ir duomenų, kad tai darytų neigiamą įtaką mokymuisi. Tačiau visuotinai pripažįstama, kad žinios apie besimokančiųjų mokymosi stilių padeda planuoti ir įgyvendinti į studentus orientuotą ugdymą (Fleming, 2001; Stash, Cristea, Bra, 2004).

Pastaraisiais dešimtmečiais informatikos inžinerijos mokslinėje literatūroje pagausėjo tyrimų, susijusių su mokymosi stilių teorijomis. Jos pradedamos įgyvendinti ir praktiškai: informatikos inžinerijoje sprendžiami šiuolaikiškų adaptyvių ar rekomendacinių mokymosi sistemų projektavimo uždaviniai. Tokio tipo mokymosi sistemos leistų organizuoti mokymąsi atsižvelgiant į besimokančiųjų mokymosi stilius (Preidys ir kt., 2010; 2012; Urbonienė, 2012; 2013; Deborah ir kt., 2014; Kurilovas ir kt., 2014 ir kt.). Akivaizdu, kad projektuojant sistemas derėtų rinktis edukologiniais empiriniais tyrimais patikrintas mokymosi stilių tipologijas. Darnus edukologijos, informatikos inžinerijos ir psichologijos kryptių mokslininkų darbas sudarytų prielaidas sėkmingai įgyvendinti šiuolaikiškas adaptyvias ar rekomendacines mokymosi sistemas.

Mokymosi stiliaus ir mokymosi metodo dermę nusako mokymosi stiliaus ypatybių ir mokymosi metodo suteikiamų galimybių tarpusavio atitiktis – tai suteikia galimybę labiau prisiderinti prie besimokančiojo poreikių. Teorinė literatūros šaltinių apie mokymosi stilių ir mokymosi metodų dermę analizė parodė, kad iki šiol atliekant pradedančiųjų programavimo gebėjimų ugdymosi pagal mokymosi stilius tyrimus paprastai akcentuojama tai, kad skirtingi individai mokosi skirtingai ir šie skirtumai lemia jų mokymosi rezultatus. Pavyzdžiui, pradedančiųjų programavimo gebėjimų ugdymąsi, atsižvelgiant į skirtingus informacijos priėmimo būdus (pagal R. M. Felderio ir L. K. Silverman mokymosi stilių tipologiją), tyrinėjo mokslininkų grupė (Thomas ir

kt., 2002) Jungtinėje Karalystėje. Mokslininkai nustatė statistiškai reikšmingą skirtumą, įrodantį, kad mokymosi rezultatai priklauso nuo besimokančiųjų mokymosi stiliaus.

Vis dėlto pradedančiųjų programavimo gebėjimams ugdytis taikomų mokymosi metodų tyrimai dažniau siejami su programavimo gebėjimams ugdytis pasitelkiamomis vizualiojo programavimo aplinkomis ir jų tyrimu (Bennedsen, Caspersen, 2004; Sorva, 2012), o rečiau – su mokymosi stilių tipologijomis. Kiek išsamiau tyrinėta grupinio darbo metodų ir mokymosi stilių dermė (Huxham, Land, 2000). Nustatyta, kad, formuojant subalansuotas besimokančiųjų grupes (t. y. į grupę įtraukiami visų mokymosi stilių atstovai), darbas grupėse vyksta sklandžiau, pasiekiami geresni rezultatai (Huxham, Land, 2000). Atliekant disertacinį tyrimą buvo taikomi darbo grupėse metodai, o formuojant grupes buvo laikomasi šių rekomendacijų. Tad taikant metodą pastebėta, kad jei tik padaugėja abejonių dėl užduoties sprendimo būdo pasirinkimo ir atsiranda poreikis dalytis darbą, darbo sėkmę visuomet nulemia gerai suformuota grupė. Be abejonės, verta išsamiau tyrinėti mokymosi stiliaus įtaką dirbant grupėse, kai sprendžiamos sudėtingesnės programavimo užduotys, mat jeigu subalansuotose grupėse pasiekiami geresni rezultatai, rekomendacijas galima būtų teikti ir įmonėms, formuojančioms programuotojų komandas. Tačiau tokį tyrimą reikėtų atlikti analizuojant aukštesniųjų kursų studentų mokymosi stilių ir programavimo gebėjimų dermę.

Šiame darbe atliktu MSMMD modelio tyrimu nustatyta mokymosi stilių ir mokymosi metodų dermės raiška pradedančiųjų programavimo gebėjimų ugdymosi procese. Tyrimu atskleisti taikytų mokymosi metodų ypatumai sudaro prielaidas tobulinti pradedančiųjų programavimo gebėjimų ugdymosi procesą. Taip pat nustatyta, kad pasirinkta mokymosi stilių tipologija netinkama norint įvertinti algoritmų vizualizavimo metodo tinkamumą skirtingiems mokymosi stiliams. Vadinasi, siekiant išsiaiškinti mokymosi stilių ir mokymosi metodų dermės poveikį, būtina atlikti lyginamuosius tyrimus, naudojant kitas mokymosi stilių tipologijas. Todėl, atsižvelgiant į besimokančiųjų mokymosi metodų taikymo vertinimus ir apibūdinimus, teoriniu pagrindu sudarytas MSMMD modelis, skirtas pradedančiųjų programavimo gebėjimams ugdytis, pakoreguotas – tai išskirtinė veiklos tyrimo charakteristika.

Atlikdami tyrimą dalyviai ugdėsi gebėjimą pažinti save ir įsivertinti turimą mokymosi stilių, tobulino mokėjimo mokytis kompetenciją. Žinojimas, kad dėstytojas mokymosi veiklas parenka ir organizuoja taip, kad atitiktų studentų turimą patirtį, gebėjimus, mokymosi stilių, gerina besimokančiųjų psichologinę savijautą ir stiprina mokymosi motyvaciją. Tyrimo metu sukaupta informacija apie mokymosi stilių taikymą pedagoginėje praktikoje ir sudaryta tyrimo metodologija gali būti naudinga tolimesniems tokio pobūdžio tyrimams.

IŠVADOS

1. Remiantis mokymosi stilių analize buvo išskirta keletas mokymosi stilių taikymo sričių: savęs pažinimui (savianalizei), mokėjimo mokytis kompetencijai ugdyti, mokymuisi individualizuoti ir diferencijuoti, profesinio mokymosi ir karjeros kryptčiai nustatyti, konsultuoti (bendravimo, karjeros, mokymosi, personalo pasirinkimo, šeimos klausimais), grupiniams procesams organizuoti (komandoms kurti, personalo bendravimui ir bendradarbiavimui gerinti, konfliktams analizuoti ir valdyti), adaptyvioms ir rekomendacinėms sistemoms projektuoti.
2. Sklandžią pradedančiųjų programavimo proceso eigą lemia loginio mąstymo, problemų sprendimo, programavimo konceptų suvokimo, programavimo kalbos sintaksės ir semantikos, informacijos paieškos ir analizės, programinių fragmentų derinimo, programavimo aplinkos naudojimo, programavimo technikos ir metodų taikymo gebėjimų visuma, kuri ir gali būti įvardijama kaip pradedančiųjų programavimo gebėjimas.
3. Pradedančiųjų programavimo gebėjimų ugdymosi teorinis tyrimas atskleidė, kad įvertinus mokymosi stilių ir mokymosi metodų dermės veiksnį galima greičiau įveikti programavimo mokymosi pradžioje kylančius sunkumus, susijusius su mokymosi metodų pasirinkimu ir taikymu. Sukurtas mokymosi stilių ir mokymosi metodų dermės modelis tinka rengiant mokymosi stilių atitinkančius mokymosi scenarijus (pavyzdžiui, planuojant darbą grupėse, numatant mokymosi stilių atitinkančias mokymosi veiklas ir pan.). Modelis taip pat taikytinas kuriant adaptyvias ar rekomendacines programavimo mokymosi aplinkas, kai besimokančiajam mokymosi veiklas parenkamos automatizuotai, ir ugdant besimokančiųjų mokėjimo mokytis kompetenciją. Siekiant veiksmingai taikyti modelį rekomenduotina supažindinti studentus su mokymosi metodų pasirinkimo strategija.
4. Žvalgomojo tyrimo duomenų analizė atskleidė, kad:
 - aukštųjų mokyklų įvadinųjų programavimo kursų aprašuose kaip dalyko tikslas dažniausiai nurodomas programavimo konceptų supratimo ugdymas ir programavimo kalbos sintaksės žinių suteikimas. Programavimo užduočių analizės, problemų sprendimo, programinių fragmentų derinimo ir kt. svarbių gebėjimų ugdymas minimas retai;

-
- aukštosiose mokyklose (universitetuose ir kolegijose) ugdant pradedančiųjų programavimo gebėjimus iš esmės vyrauja mokymo paradigmą atitinkantis studijų pobūdis: studentai klauso paskaitų, sprendžia individualias programavimo užduotis, konsultuojasi su dėstytojais ir bendramoksliais. Remiantis dokumentų analize, galima būtų teigti, kad kolegijose dažniau taikomi mokymosi paradigmą atitinkantys aktyvaus darbo studijų metodai (darbas grupėse, diskusijos, minčių lietus, probleminis dėstymas ir kt.), tačiau kiekybinio anketinio tyrimo duomenys šios prielaidos nepatvirtina – esminio skirtumo tarp kolegijų ir universitetų praktikoje taikytų programavimo mokymosi metodų nenustatyta.
5. Kokybinio tyrimo duomenų analizė atskleidė, kad:
- dažniausiai studentų nuomonę apie programavimo mokymosi metodų tinkamumą formuoja veiksniai, susiję su besimokančiojo poreikiais ir mokymosi metodo teikiamomis galimybėmis. Mokymosi stiliaus ir mokymosi metodo dermės veiksnys identifiкуotas retai;
 - tyrimo duomenys suponuoja prielaidą, kad algoritmų vizualizavimo metodo atitikimas mokymosi stiliams pagal P. Honey ir A. Mumfordo mokymosi stilių tipologiją nėra pagrįstas – reikalingi papildomi išsamesni tyrimai. Šį metodą iš esmės apibūdina vizualus informacijos perteikimas, o P. Honey ir A. Mumfordo mokymosi stilių tipologijoje mokymosi stiliai neskirstomi pagal vizualios / nevizualios informacijos priėmimą;
 - teorinės analizės pagrindu sudarytas mokymosi stilių ir mokymosi metodų dermės modelis neleido visapusiškai įvertinti diskusijų, programos teksto skaitymo ir veikimo sekimo metodų teikiamų galimybių: 1) pasyvaus dalyvavimo diskusijose galimybės tinkamumo teoretiko mokymosi stiliui ir 2) išsamiai, žingsnis po žingsnio, aiškintis programos galimybės tinkamumą mąstytojo mokymosi stiliui, taikant programos teksto skaitymo ir veikimo sekimo metodą. Remiantis tyrimo rezultatais mokymosi stilių ir mokymosi metodų dermės modelis patobulintas priskiriant mінėtus mokymosi metodus atitinkamiems mokymosi stiliams.

PASIŪLYMAI IR PRAKTINĖS REKOMENDACIJOS

Tyrimo išvadų pagrindu suformuluoti šie pasiūlymai ir praktinės rekomendacijos:

Universitetų ir kolegijų bendruomenėms:

- Reikalingi lyginamieji tyrimai, kurie leistų išryškinti mokymosi stilių ir mokymosi metodų dermės poveikį, naudojant kitas mokymosi stilių tipologijas, kurios apimtų ne tik besimokantiesiems būdingas mokymosi veiklas, bet ir skirtingus informacijos priėmimo būdus.
- Rekomenduojama taikyti sudarytą mokymosi stilių ir mokymosi metodų dermės modelį projektuojant adaptyvias ar rekomendacines sistemas, skirtas programavimo gebėjimams ugdytis.
- Esant galimybei, įvadinio programavimo kursą dėstyti išimtinai kompiuterių laboratorijose, neskiriant atskirai laiko auditorinių paskaitų skaitymui, t. y. naują mokymosi medžiagą pateikti pasitelkiant informacines komunikacines technologijas, o kontaktinį laiką naudoti praktiniams gebėjimams ugdyti ir konsultacijoms teikti.

Įvadinių programavimo kursų dėstytojams:

- Ugdant pradedančiųjų programavimo gebėjimus reikėtų atsižvelgti į visus programavimo proceso etapus: užduoties analizę, sprendimo algoritmo sudarymą, programos teksto rašymą, programos derinimą ir testavimą, o programas mokyti kurti vadovaujantis anksčiau išvardytais etapais.

NAUDOTA LITERATŪRA

1. Adomaitienė, J., Zubrickienė, I., Teresevičienė, M. (2008). Profesinis mokytojų ir dėstytojų tobulėjimas, atliekant veiklos tyrimus. Profesinis rengimas. *Tyrimai ir realijos*, 15, 10–21.
2. *Aktyvaus mokymosi metodai: Mokytojo knyga*. (1998). Vilnius: Garnelis.
3. Allinson, C. W., Hayes, J. (1988). The learning styles questionnaire: An alternative to Kolb's inventory. *Journal of Management Studies*, 25 (3), 269–281.
4. Anderson, L., Krathwohl, D. (2001). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Addison-Wesley.
5. Aronson, E. (2000). *Nobody left to hate: Teaching compassion after Columbine*. New York: W. H. Freeman. [žiūrėta 2013 m. sausio 15 d.]. Prieiga per internetą: <http://www.jigsaw.org>.
6. Argyris, C., Schön, D. A. (1974). *Theory in Practice: Increasing Professional Effectiveness*. San Francisco: Jossey-Bass.
7. Atkinson, G. A. (1991). Kolb's Learning-Style Inventory: A practitioner's perspective. *Measurement & Evaluation in Counseling & Development*, 23, 149–161. Available: EBSCOhost Full Display (2001, June 1).
8. Ausubel, D. (2000). *The Acquisition and Retention of Knowledge: A Cognitive View*. Springer, 232.
9. Bal, H. E., Grune, D. (1994). *Programming language Essentials*. Addison-Wesley.
10. Balevičienė, S., Paurienė, V. (2013) Mokėjimo mokytis kompetencija: sampratos ir ugdymas. *Švietimo problemas analizė*, 16 (102). Vilnius: LR švietimo ir mokslo ministerijos Švietimo aprūpinimo centras. [žiūrėta 2014 m. birželio 10 d.]. Prieiga per internetą: http://www.smm.lt/uploads/lawacts/docs/575_d626f461a75e8801d33751d38b078dae.pdf.
11. Barkauskaitė, M., Motiejūnienė, E. (2004). Mokymosi motyvacijos problema ir jos sprendimo galimybės. *Pedagogika*, 70, 38–43.
12. Barnes, D. J., Fincher, S., Thompson, S. (1997). Introductory problem solving in computer science. In G. Daughton and Magee (eds.). *5th Annual Conference on the Teaching of Computing* (pp. 36–39). Newtownabbey, UK: HE Academy for Information and Computer Sciences.
13. Barnes, D. J., Kölling, M. (2002). *Objects first with Java – A practical introduction using BlueJ*. Hemel Hempstead, UK: Prentice Hall/Pearson Education.
14. Ben-Ari, M. (2001). Program visualisation in theory and practice. *The European Online Magazine for the IT Professional*, II (2), 8–11. [žiūrėta 2012 m. balandžio 15 d.]. Prieiga per internetą: <http://www.upgrade-cepis.org>.
15. *Bendrieji Europos kalbų mokymosi, mokymo ir vertinimo metmenys*. (2008). Vilnius: Firidas.
16. Bennane, A. (2013). Adaptive educational software by applying reinforcement learning. *Informatics in Education*, 12 (1), 13–28.
17. Bennedsen, J., Caspersen, M. E. (2004) Programming in Context – A Model-First Approach to CS1. *SIGCSE BULLETIN*, 36, 1, 477–481.
18. Bennedsen, J., Caspersen, M. E. (2008) Optimists have more fun, but do they learn better? On the influence of emotional and social factors on learning introductory computer science. *Computer Science Education*, 18 (1), 1–16.
19. Bennedsen, J. (2008). *Teaching and Learning Introductory Programming – A Model-Based Approach*. Dissertation for Dr. Philos degree in the Faculty of mathematics and Natural Sciences, University of Oslo, Norway.

20. Beres, I., Maguar, T., Turcsanyj-Szabo, M. (2012). Towards a personalised, learning style based collaborative blended learning model with individual assessment. *Informatics in Education*, 11 (1), 1–28.
21. Berglund, A., Lister, R. (2010). Introductory Programming and the Didactic Triangle. In *Proceedings of 12th Australasian Computing Education Conference (ACE 2010)*. Brisbane, Australia.
22. Berry, M. *The new computing curriculum*. [žiūrėta 2014 m. gegužės 3 d.]. Prieiga per internetą: <http://milesberry.net/2013/02/the-new-computing-curriculum-some-thoughts/>.
23. Biggs, J., Tang, C. (2007). *Teaching for quality learning at university* (3rd Ed). Society for Research into Higher Education & Open University Press, 357.
24. Biggs, J. B. (2003). *Teaching for quality learning at university*. Buckingham: Open University Press / Society for Research into Higher Education. (Second edition).
25. Bitinas, B. (2002). *Pedagoginės diagnostikos pagrindai*. Vilnius: Parama.
26. Bitinas, B., Rupšienė, L., Žydžiūnaitė, V. (2008). *Kokybinių tyrimų metodologija*. Klaipėda.
27. Bitinas, B. (2000). *Ugdymo filosofija*. Vilnius: Enciklopedija.
28. Bitinas, B. (2006). *Edukologinis tyrimas: sistema ir procesas*. Vilnius: Kronta.
29. Bitinas, B. (2013). *Rinktiniai edukologiniai raštai*. Vilnius: Edukologija.
30. Blocher, M. J. (2005). Increasing learner interaction: using Jigsaw online. *Educational Media International*, 42, 3, 269–278.
31. Bloom, B. S., Engelhart, M. D., Furst, E. J, Hill, W., Krathwohl, D. (1956). *Taxonomy of educational objectives. Vol. I: The cognitive domain*. New York: McKay.
32. Bloom, B. S., Masia, B. B., Krathwohl, D. R. (1964). *Taxonomy of Educational Objectives. Vol. II: The Affective Domain*. New York: McKay.
33. Bloun, T. (2003). *Felder-Silverman Learning Styles Model*. [žiūrėta 2009 m. gruodžio 10 d.]. Prieiga per internetą: <http://chat.carleton.ca/~tblouin/Felder/felder.html>.
34. Brew, A. (2003). *Towards Autonomous Assessment: Using Self Assessment and Peer Assessment. Assessment Matters in Higher Education*. Buckingham: SRHE and Open University Press, 159–171.
35. Bruce, K. (2004). Controversy on How to Teach CS1: A Discussion on the SIGCSE-members Mailing List. In *inroads – The SIGCSE Bulletin*, December. [žiūrėta 2014 m. sausio 20 d.]. Prieiga per internetą: <http://www.cs.pomona.edu/~kim/ftp/Inroads.pdf>.
36. Brooks, J. G. (1999). *In search of understanding: The case for constructivist classrooms*. ASCD. [žiūrėta 2015 m. sausio 18 d.]. Prieiga per internetą: https://www.google.lt/books?hl=lt&lr=&id=9W_VB5TjxxoC&oi=fnd&pg=PR7&ots=3HHClrXEj1&sig=MOWpgGROCS56NjxX2o01n1VoN_s&redir_esc=y#v=onepage&q&f=false.
37. Bruzdelevičienė, R. (2008). Ugdymo paradigmu kaitos permanentiškumo problema: teorija – praktinė veikla – mokytojų rengimas. *Mokytojų ugdymas: mokslo darbai*, 10, 74–90.
38. Bubnys, R., Žydžiūnaitė, V. (2012). *Reflektyvusis mokymas(is) aukštosios mokyklos edukacijoje aplinkoje: dėstytojų mokymo patirtys: mokslo studija*. Šiauliai: Lucilijus.
39. Bulajeva, T., Lepaitė, D., Šileikaitė-Kaishauri, D. (2011). *Studijų programų vadovas: metodinė priemonė studijų programų komitetų nariams ir dėstytojams*. Vilnius: Kriventa.
40. Butkienė, G., Kepalaitė, A. (1996). *Mokymasis ir asmenybės brendimas*. Vilnius: Margi raštai.
41. Caspersen, M. E. (2007). *Educating Novices in The Skills of Programming*. PhD Dissertation, Department of Computer Science University of Aarhus, Denmark.

42. Cassidy, S. (2004). Learning Styles: An overview of theories, models, and measures. *Educational Psychology*, 24 (2), 419–444. [žiūrėta 2014 m. balandžio 15 d.]. Prieiga per internetą: http://www.acdowd-designs.com/sfsu_860_11/LS_OverView.pdf.
43. CEDEFOP. (2008). *The shift to learning outcomes*. [žiūrėta 2014 m. liepos 4 d.]. Prieiga per internetą: http://www.cedefop.europa.eu/EN/Files/3054_en.pdf.
44. Chislet, V., Chapman, A. (2005). *VAK learning styles self-test*. [žiūrėta 2012 m. spalio 9 d.]. Prieiga per internetą: <http://www.businessballs.com.vaklearningstylestest.htm>.
45. Churches, A. (2009). *Bloom's Digital Taxonomy*. [žiūrėta 2013 m. gegužės 10 d.]. Prieiga per internetą: <http://edorigami.wikispaces.com/file/view/bloom%27s%20Digital%20taxonomy%20v3.01.pdf>.
46. Coffield, F. J., Moseley, D. V., Hall, E., Ecclestone, K. (2004). *Learning styles and pedagogy in post-16 learning: a systematic and critical review*. London: Learning and Skills Research Centre/University of Newcastle upon Tyne. [žiūrėta 2014 m. balandžio 10 d.]. Prieiga per internetą: <http://www.lsd.org.uk/files/pdf/1543.pdf>.
47. *Computer Science Curriculum 2013*. (2013). [žiūrėta 2014 m. kovo 23 d.]. Prieiga per internetą: <http://ai.stanford.edu/users/sahami/CS2013/ironman-draft/cs2013-ironman-v1.0.pdf>.
48. Cox, S. M., Tsai, K. C. (2013). Exploratory Examination of Relationships between Learning Styles and Learner Satisfaction in Different Course Delivery Types. *International Journal of Social Science Research*, 1 (1), 64–76. [žiūrėta 2014 m. liepos 3 d.]. Prieiga per internetą: <http://www.macrothink.org/journal/index.php/ijssr/article/viewFile/4100/3388>.
49. Curry, L. (1990). One critique of the research on learning styles. *Educational Leadership*, 48, 50–56.
50. Cuthbert, P. F. (2005). The student learning process: Learning styles or learning approaches? *Teaching in Higher Education*, 10 (2), 235–249.
51. Čekanavičius, V., Murauskas, G. (2003). *Statistika ir jos taikymai I*. Vilnius: TEV.
52. Dagdilelis, V., Satratzemi, M., Evangelidis, G. (2004). Introducing secondary education to algorithms and programming. *Education and Information Technologies*, 9 (2), 159–173.
53. Dagienė, V., Grigas, G. (2007). *Programavimo kalbų teoriniai pagrindai*. [žiūrėta 2014 m. rugsėjo 15 d.]. Prieiga per internetą: <http://ims.mii.lt/valentina/publ/Progrkalb.pdf>.
54. Dagienė, V., Grigas, G., Jevsikova, T. (2008). *Enciklopedinis kompiuterijos žodynas*. Vilnius: TEV.
55. Dagienė, V., Urbonienė, J. (2010). Learning Programming: Comparative Analysis of Languages and Environments. *Information Sciences*, 54, 44–62.
56. Dagienė, V., Žilinskienė, I. (2011). Mokymosi veiklų samprata skaitmeninėje erdvėje. *Pedagogika*, 102, 94–103.
57. Davies, R., Dean, D., Ball, N. (2013). Flipping the classroom and instructional technology integration in a college-level information systems spreadsheet course. *Educational Technology Research and Development*, 61 (4).
58. Deborah, L. J., Baskaran, R., Kannan, A. (2014). Learning styles assessment and theoretical origin in an E-learning scenario: a survey. *Artificial Intelligence Review*, 42 (4), 801–819.
59. DeCiantis, S. M., Kirton, M. J. (1996). A Psychometric Re-examination of Kolb's Experiential Learning Cycle Construct: A Separation of Level, Style and Process. *Educational and Psychological Measurement*, 56, 5, 809–820.
60. De Raadt, M., Watson, R., Toleman, M. (2003). Introductory programming languages at Australian universities at the beginning of the twenty first century. *Journal of Research and Practice in Information Technology*, 3, 163–167.

61. Dewey, J. (2004). *Democracy and education*. Courier Corporation. [žiūrėta 2015 m. sausio 18 d.]. Prieiga per internetą: <https://www.google.lt/books?hl=lt&lr=&id=19ajcXf4MCCY&oi=fnd&pg=PA1&dq=Democracy+and+Education+by+John+Dewey>.
62. Dewey, J. (2013). *1859–1952. Demokratija ir ugdymas: įvadas į ugdymo filosofiją*. Klaipėda: Baltic Printing House.
63. Dorca, F. A., Lima, L. V., Fernandes, M. A., Lopes, C. A. (2012). A stochastic approach for automatic and dynamic modeling of students' learning styles in adaptive educational systems. *Informatics in Education, 11* (2), 191–212.
64. Du Boulay, B. (1989). Some difficulties of learning to program. In E. Soloway and J. C. Spohrer (Eds.). *Studying the novice programmer*. Hillsdale, NJ: Lawrence Erlbaum, 283–299.
65. Duff, A., Duffy, T. (2002). Psychometric properties of Honey & Mumford's Learning Styles Questionnaire (LSQ). *Personality and individual differences, 33* (1), 147–163.
66. Dunn, R., Griggs, S. A. (2000). *Practical Approaches to Using Learning Styles in Higher Education*. Westport, Connecticut: Bergin & Garvey.
67. Dunn, R., Griggs, S. A., Olson, J., Beasley, M., Gorman, B. S. (1995). A meta-analytic validation of the Dunn and Dunn model of learning style preference. *Journal of Educational Research, 88*, 6, 353–361.
68. Džeimsas, V. (1995). *Pragmatizmas*. Vilnius: Pradai.
69. Entwistle, N. (1987). A model of the teaching-learning process. In T. E. Richardson, M. W. Eysenck & D. W. Piper. *Student Learning: Research in Education and Cognitive Psychology*. Milton Keynes: SRHE and OUP, 13–28.
70. Entwistle, N. J., Peterson, E. R. (2004). Conceptions of learning and knowledge in higher education: Relationships with study behaviour and influences of learning environments. *International Journal of Educational Research, 41* (6), 407–428.
71. Exton, C. (2002). Constructivism and program comprehension strategies. In *Proceedings of the 10th International Workshop on Program Comprehension (IWPC'02)* (pp. 281–284). La Sorbonne, Paris, France, June 26–29.
72. *Europos Parlamento ir Tarybos rekomendacija dėl bendrųjų visą gyvenimą trunkančio mokymosi gebėjimų*. (2006). [žiūrėta 2015 m. sausio 20 d.]. Prieiga per internetą: <http://eur-lex.europa.eu/legal-content/LT/TXT/?uri=CELEX:32006H0962>.
73. Factorovich, P. (2006). Evolution of computer science degrees at science school. In *19th IFIP World Computer Congress-WCC 2006*.
74. Farag, W., Ali, S., Deb, D. (2013). Does language choice influence the effectiveness of online introductory programming courses? In *Proceedings of the 13th annual ACM SIGITE conference on Information technology education* (pp. 165–170). ACM.
75. Felder, R. M., Silverman, L. K. (1988). Learning and teaching styles in engineering education. *Engineering education, 78* (7), 674–681.
76. Felder, R. M., Brent, R. (2005). Understanding Student Differences. *J. Engr. Education, 94* (1), 57–72. [žiūrėta 2013 m. sausio 20 d.]. Prieiga per internetą: http://www4.ncsu.edu/unity/lockers/users/f/felder/public/Papers/Understanding_Differences.pdf.
77. Felder, R. M., Spurlin, J. (2005). Applications, Reliability and Validity of the Index of Learning Styles. *Intl. Journal of Engineering Education, 21* (1), 103–112. [žiūrėta 2013 m. sausio 18 d.]. Prieiga per internetą: [https://wss.apan.org/jko/mls/Learning%20Content/ILS_Validation\(IJEE\).pdf](https://wss.apan.org/jko/mls/Learning%20Content/ILS_Validation(IJEE).pdf).

78. Fleming, N. D. (2001). *Teaching and learning styles: VARK strategies*. Christchurch, New Zealand: N.D. Fleming.
79. Fleming, S., Mckee, G. & Huntley-Moore, S. (2011). Undergraduate nursing students' learning styles: a longitudinal study. *Nurse education today*, 31 (5), 444–449.
80. Frand, J. L. (2006). The Information Mindset: Changes in Students and Implications for Higher Education. *EDUCAUSE Review* 41 (March / April), 14–16.
81. Gage, N. J., Berliner, D. C. (1993). *Pedagoginė psichologija*. Vilnius: Alna litera.
82. Gardner, H. (1983). *Frames of mind. The Theory of Multiple Intelligence. Basic Books*. [žiūrėta 2011 m. spalio 21 d.]. Prieiga per internetą: <http://www.scribd.com/doc/68879364/Frames-of-Mind-The-Theory-of-Multiple-Intelligences>.
83. Garnham, C., Kaleta, R. (2002). Introduction to Hybrid Courses. *Teaching with Technology Today*, 8, 6. [žiūrėta 2009 m. kovo 4 d.]. Prieiga per internetą: <http://www.uwsa.edu/ttt/articles/garnham.htm>.
84. Gibbs, D. C. (2000). The effect of a constructivist learning environment for field-dependent/independent students on achievement in introductory computer programming. In *SIGCSE Bulletin*, 03/00 (pp. 207–211). Austin, Texas, USA.
85. Gliem, J. A., Gliem, R. R. (2003). Calculating, interpreting, and reporting Cronbach's alpha reliability coefficient for Likert-type scales. In *Midwest Research-to-Practice Conference in Adult, Continuing, and Community Education*.
86. Glosienė, A. (2006). Akademinės bendruomenės informacinės kompetencijos ugdymas: po dvidešimties metų. Iš *Knygotyra*. [žiūrėta 2015 m. sausio 21 d.]. Prieiga per internetą: <http://etalpykla.lituanistikadb.lt/fedora/objects/LT-LDB-0001:J.04~2006~1367153999037/datastreams/DS.002.0.01.ARTIC/content>.
87. Gomes, A., Mendes, A. J. (2007). Learning to program – difficulties and solutions. In *International Conference on Engineering Education – ICEE 2007*. Coimbra, Portugal.
88. Gomes, A., Carmo, L., Bigotte, E., Mendes, A. (2006). Mathematics and programming problem solving. In *Proc. of the 3rd E-Learning Conf. – Computer Science Education*. September, Coimbra, Portugal.
89. Gonzales, G. (2004). Constructivism in an introduction to programming course. *JCSC*, 19 (4), 299–305.
90. Grandy, R. E. (2010). *Constructivisms, Scientific Methods, and Reflective Judgment in Science Education*. The Oxford Handbook of Philosophy of Education. Ed. Siegel Harvey. Oxford Handbooks Online. January. [žiūrėta 2015 m. sausio 18 d.]. Oxford handbooks online doi:10.1093/oxfordhb/9780195312881.003.0021.
91. Gudžinskienė, V. (2008). Mokymo ir mokymosi sampratų analizė. *Pedagogika*, 90, 49–56.
92. Guo, P. J. (2013). Online Python Tutor: Embeddable web-based program visualization for CS education. In *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 579–584). ACM.
93. Hadjerrouit, S. (2008). Towards a Blended Learning Model for Teaching and Learning Computer Programming: A Case Study. *Informatics in Education*, 7, 2, 181–210.
94. Hadjerrouit, S. (1998). Java as first programming language: a critical evaluation. *SIGCSE Bull.* 30, 2 (June 1998), 43–47. [žiūrėta 2015 m. sausio 19 d.]. <http://doi.acm.org/10.1145/292422.292440>.
95. Hawk, T. F., Shah, A. J. (2007). Using Learning Style Instruments to Enhance Student Learning. *Decision Sciences Journal of Innovative Education*, 5, 1, January, U.S.A., 1–19.

96. Henard, F., Roseveare, D. (2012). *An IMHE Guide for Higher education Institutions*. [žiūrėta 2014 m. birželio 18 d.]. Prieiga per internetą: <http://www.oecd.org/edu/imhe/QT%20policies%20and%20practices.pdf>.
97. Hendry, G. D., Heinrich, P., Lyon, P. M., Barratt, A. L., Simpson, J. M., Hyde, S. J., Mgiaeth, S. (2005). Helping students understand their learning styles: Effects on study self-efficacy, preference for group work, and group climate. *Educational Psychology*, 25 (4), 395–407.
98. Hennessy, D. (2011). *Learning Style, Teaching Style, and High School Student Retention*. [žiūrėta 2014 m. balandžio 20 d.]. Prieiga per internetą: <http://search.proquest.com/docview/862946314>.
99. Heron, J. (1999). *The Complete Facilitator's Handbook*. London: KoganPage.
100. Holmboe, C. (2005). *Language, and the learning of data modelling*. [žiūrėta 2015 m. sausio 19 d.]. Prieiga per internetą: http://folk.uio.no/christho/phd/thesis_Holmboe.pdfSaid.
101. Honey, P. (2006). *Learning Styles Questionnaire (40-Item)*. Maidenhead Berks: Peter Honey Publication Limited.
102. Honey, P., Mumford, A. (1992). *The manual of learning styles*. Maidenhead: Peter Honey.
103. Honey, P., Mumford, A. (2006). *Learning styles questionnaire*. 80 items version.
104. Kennedy, D., Hyland, A., Ryan, N. (2009). *Learning outcomes and competences*. Bologna Handbook, Introducing Bologna Objectives and Tools.
105. Huxham, M. & Land, R. (2000). Assigning students in group work projects. Can we do better than random? *Innovations in Education and Teaching International*, 37 (1), 17–22.
106. Ignatova, N., Kurilovas, E. (2012). Informacinėmis ir komunikacinėmis technologijomis grįsto mokymo ir mokymosi individualizavimo kryptys Lietuvos švietimo kontekste. *Pedagogika*, 26, 21–29.
107. Yong, C. F. (2007). *Automatically generating adaptive programming exercises based on student profiles*. Thesis for the degree of Doctor of Philosophy, the University of Nottingham.
108. Jadzgevičienė, V., Urbonienė, J. (2013). The Possibilities of Virtual Learning Environment Tool Usability for Programming Training. In *Proceedings of the 6th International Conference Innovative Information Technologies for Science, Business and Education*, IIT-2013 November, 14–16.
109. Jasiūnaitė, B. (2004). *Edukacinės novacijos ir jų diegimas*. Kaunas: Technologija.
110. Jenkins, T. (2002). On the Difficulty of Learning to Program. In *Proceedings of 3rd annual conference of the LTSN-ICS* (pp. 53–58). Loughborough University, United Kingdom, August 2002.
111. Jensen, E. (2001). *Tobulus mokymas: daugiau kaip 1000 praktinių patarimų vaikų ir suaugusiųjų mokytojams*. Vilnius: AB OVO.
112. Jovaiša, L. (2002). *Edukologijos įvadas*. Vilnius: Vilniaus universiteto leidykla.
113. Jovaiša, L. (2007). *Enciklopedinis edukologijos terminų žodynas*. Vilnius: Gimtasis žodis.
114. Juodaitytė, A. (2012). *Kokybiniai edukaciniai tyrimai: teorijos, duomenų rinkimas ir analizė*. Šiauliai: VšĮ Šiaulių universiteto leidykla.
115. Juozaitis, A. M., Juozaitienė, R. (2003). *Savarankiško mokymosi gidas*. Vilnius: Inter Se.
116. Juškevičienė, A., Urbonienė, J. (2011). The Research on the Web 2.0 Technologies Adaptation for Teaching Programming. *Studies in Modern Society*, 2 (1), 161–170.
117. Kappe, F. R., Boekholt, L., Den Rooyen, C. & Van der Flier, H. (2009). A predictive validity study of the Learning Style Questionnaire (LSQ) using multiple, specific learning criteria. *Learning and Individual Differences*, 19 (4), 464–467.

118. Kardelis, K. (2007). *Mokslinių tyrimų metodologija*. Kaunas: Judex.
119. Kemmis, S., McTaggart, R. (2005). Participatory Action Research: Communicative Action and the Public Sphere. In K. Denzin, Norman and S. Lincoln, Yvona (Eds.). *The SAGE Handbook of Qualitative Research*. 3rd ed. London: SAGE Publications.
120. Kinnunen, P. (2009). *Challenges of teaching and studying programming at a university of technology – Viewpoints of students, teachers and the university*. Dissertation for the degree of Doctor of Philosophy of the Faculty of Information and Natural Sciences at Helsinki University of Technology (Espoo, Finland), December.
121. Kolb, D. A. (1984). *Experiential Learning: experience as the source of learning and development*. Englewood Cliffs, NJ: Prentice-Hall.
122. Kolb, D. A., Boyatzis, R. E., Mainemelis, Ch. (2000). Experimental Learning Theory: Previous Research and New Directions. In R. J. Sternberg and L. F. Zhang (Eds.). *Perspectives on cognitive, learning, and thinking styles*. NJ: Lawrence Erlbaum. [žiūrėta 2011 m. balandžio 30 d.]. Prieiga per internetą: http://www.medizin1.klinikum.unierlangen.de/e113/e191/e1223/e1228/e989/inhalt990/erfahrungslernen_2004_ger.pdf.
123. Kolb, A. Y. (2005). *The Kolb learning style inventory–version 3.1 2005 technical specifications*. Boston, MA: Hay Resource Direct, 200. [žiūrėta 2014 m. gruodžio 20 d.]. Prieiga per internetą: http://learningfromexperience.com/media/2010/08/tech_spec_lsi.pdf.
124. Kolb, A. Y., Kolb, D. A. (2005). Learning Styles and Learning Spaces: Enhancing Experiential Learning in Higher Education. *Academy of Management Learning and Education*, 4 (2), 193–212.
125. Kolb, D. A. (1984). *Experiential learning: Experience as the source of learning and development*. Englewood Cliffs, NJ: Prentice Hall.
126. Kolikant, B. D. Y., Pollack, S. (2004). Establishing computer science norms among high school students. *Computer Science Education*, 14 (1), 21–35.
127. Kölling, M. (2012). *The problem of teaching object-oriented programming*. [žiūrėta 2013 m. balandžio 30 d.]. Prieiga per internetą: <http://www.bluej.org/papers/1999-09-JOOP2-environments.pdf>.
128. Kölling, M., Rosenberg, J. (2001). Guidelines for teaching object orientation with Java. *ACM SIGCSE Bulletin*, 33, 3, 33–36.
129. Kuncaitis, R. (2009). *Suaugusiųjų mokymo(si) priemonumas Europos Sąjungos švietimo politikos kontekste*. Daktaro disertacija: socialiniai mokslai, edukologija (07 S). Kaunas: VDU leidykla.
130. Kurilovas, E., Kubilinskiene, S., Dagiene, V. (2014). Web 3.0 – Based Personalisation of Learning Objects in Virtual Learning Environments. *Computers in Human Behavior*, 30, 654–662.
131. Laužikas, J. (1974). *Mokinių pažinimas ir mokymo diferencijavimas*. Kaunas: Šviesa.
132. Laužikas, J. (1993). *Pedagoginiai raštai*. Kaunas: Šviesa.
133. Lewis, M., Reinders, H. (2008). *Using student-centered methods with teacher-centered students, 2 nd edition*. Toronto: Pippin Publishing.
134. Lichtman, M. (2013). *Qualitative research in education: a user's guide*. 3rd ed. Los Angeles [et.]: SAGE.
135. Lipeikienė, J. (2004). Programavimas šiuolaikiniame informatikos mokyme. *Pedagogika*, 70, 121–127.
136. Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., McCartney, R., Moström, J. E., Sanders, K., Seppällä, O., Simon, B., Thomas, L. (2004). A Multi-National Study of Reading and Tracing Skills in Novice Programmers. *SIGCSE Bulletin*, December, 36, 4, 119–150.

137. Lopez, M., Whalley, J., Robbins, P., Lister, R. (2008). Relationships Between Reading, Tracing and Writing Skills in Introductory Programming. In *Proceedings of the Fourth international Workshop on Computing Education Research* (pp. 101–112).
138. Lowe, J. (2000). International examinations: the new credentialism and reproduction of advantage in a globalising world. In *Assessment in Education: principles, policy and practice*, 7. London: Routledge, 363–377.
139. Lubchak, V., Kuppenko, O., Kuzikov, B. (2012). Approach to dynamic assembling of individualized learning paths. *Informatics in Education*, 11 (2), 213–225.
140. Lukošūnienė, V. (2014). *Kvalifikaciją tobulinančių suaugusiųjų mokėjimo mokytis kompetencijos raiška ir ugdymas(is)*. Summary of the doctoral dissertation: social sciences, educational sciences (07 S). Vilnius: LEU leidykla.
141. Mayer, R. E., Anderson, R. B. (1991). Animations need narrations: An experimental test of a dual-coding hypothesis. *Journal of educational psychology*, 83 (4),
142. Mayes, J. T., Fowler, C. J. H. (1999). Learning technology and usability: a framework for understanding courseware. *Interacting with Computers*, 11, 485–497.
143. Maknickas, A. (2012). *Skriptinė programavimo kalba ironperūnis ir jos taikymas mokyme*. Alytus college, 93.
144. Mannila, L. (2007). Novices' Progress in Introductory Programming Courses. *Informatics in Education*, 6, 1, 139–152.
145. Mannila, L. (2009). *Teaching Mathematics and Programming – New Approaches with Empirical Evaluation*. Turku Centre for Computer Science.
146. Manolis, C., Burns, D. J., Assudani, R., Chinta, R. (2013). Assessing experiential learning styles: A methodological reconstruction and validation of the Kolb Learning Style Inventory. *Learning and Individual Differences*, 23, 44–52.
147. Markham, S. (2004). *Learning styles measurement: a cause for concern*. [žiūrėta 2014 m. birželio 30 d.]. Prieiga per internetą: http://cerg.infotech.monash.edu.au/techreps/learning_styles_review.pdf.
148. Martišauskienė, E. (2008). Ugdomojo mokymo retrospektyva ir dabartis (V. Rajecko „Ugdomojo mokymo“ ir konstruktyvizmo paralelės). *Pedagogika*, 89, 11–17.
149. Marton, F. (1988). Describing and improving learning, Ch. 3. In R. R. Shmeck (ed.). *Learning Strategies and Learning Styles*, Plenum Press.
150. Marzano, R. J. (2005). *Naujoji ugdymo tikslų taksonomija*. Vilnius: Žara.
151. Mason, R., Cooper, G., de Raadt, M. (2012). Trends in introductory programming courses in Australian universities: languages, environments and pedagogy. In *Proceedings of the Fourteenth Australasian Computing Education Conference*, 123 (ACE '12). Australian Computer Society, Inc., Darlinghurst, Australia, 33–42.
152. McGettrick, A., Boyle, R., Ibbett, R., Lloyd, J., Lovegrove, G., Mander, K. (2005). Grand challenges in computing: Education – a summary. *The Computer Journal*, 48 (1), 42–48.
153. McGill, T. J., Volet, S. E. (1997). A conceptual framework for analyzing students' knowledge of programming. *Journal of Research on Computing in Education*, 29, 3, 276–297.
154. McNiff, J. (2001). *Action research: Principles and action (2nd ed.)*. London: Routledge.
155. Mead, J. et al. (2006). A cognitive approach to identifying measurable milestones for programming skill acquisition. In *Proceedings of ITiCSE'06* (pp. 182–194). June 26–28, Bologna, Italy.
156. Merfeldaitė, O. (2007). *Socialinės pedagoginės pagalbos komandos veiklos modelis bendrojo lavinimo mokykloje: daktaro disertacija*. Vilnius: Vilniaus pedagoginio universiteto leidykla, 2007.

157. Milman, N. (2012). The flipped classroom strategy: What is it and how can it be used? *Distance Learning*, 9 (3), 85–87.
158. *Mokymas šiandien. Ugdymo turinio diferencijavimas pritaikant skirtingiems mokiniams.* (2005). Vilnius: Žara.
159. Moon, J. (2004). *Linking levels, Learning Outcomes and Assessment Criteria.* [žiūrėta 2010 m. birželio 30 d.]. Prieiga per internetą: http://www.bologna-bergen2005.no/EN/Bol_sem/Seminars/040701-02Edinburgh/040701-02Linking_Levels_plus_ass_crit-Moon.pdf.
160. Moreno, A., Myller, N., Sutinen, E., Ben-Ari, M. (2004). Visualizing programs with Jeliot 3. In *Proceedings of the working conference on Advanced visual interfaces* (pp. 373–376). ACM.
161. Morkevičius, V., Telešienė, A., Žvaliauskas, G. (2008). *Kompiuterizuota kokybinių duomenų analizė su NVivo ir Text Analysis Suite.* [žiūrėta 2013 m. liepos 3 d.]. Prieiga per internetą: http://www.lidata.eu/files/mokymai/NVivo/KKDA_20080914_esfui.pdf.
162. Noffke, S. E., Somekh, B. (2009). *The SAGE handbook of educational action research.* Thousand Oaks, Sage Publications.
163. O’Kelly, J., Mooney, A., Ghent, J., Gaughran, P., Dunne, S., Bergin, S. *An Overview of the Integration of Problem Based Learning into an existing Computer Science Programming Module.* [žiūrėta 2012 m. kovo 10 d.]. Prieiga per internetą: eprints.nuim.ie/726/1/PBLPaper2.pdf.
164. Pashler, H., McDaniel, M., Rohrer, D., Bjork, R. (2008). Learning styles: Concepts and evidence. *Psychological Science in the Public Interest*, 9, 105–119.
165. Penger, S. (2009). Testing Dunn & Dunn’s and Honey & Mumford’s learning style theories: The case of the Slovenian higher education system. *Management: Journal of Contemporary Management Issues*, 14 (2), 1–20.
166. Peterson, E. R., Rayner, S. G., Armstrong, S. J. (2009). Herding cats: In search of definitions of cognitive styles and learning styles. *ELSIN Newsletter, an international forum*, Winter 2008–2009. [žiūrėta 2011 m. gegužės 10 d.]. Prieiga per internetą: <http://www.elsinnews.com>.
167. Petty, G. (2007). *Šiuolaikinis mokymas.* Vilnius: Tyto Alba.
168. Petty, G. (2008). *Įrodymais pagrįstas mokymas.* Vilnius: Tyto Alba.
169. Petty, G. *Improve your teaching and that of your team.* [žiūrėta 2014 m. kovo 4 d.]. Prieiga per internetą: <http://geoffpetty.com>.
170. Philip, J. G. (2013). Online Python Tutor: Embeddable Web-Based Program Visualization for CS Education. In *Proceedings of the ACM Technical Symposium on Computer Science Education (SIGCSE).*
171. Pickworth, G. E., Schoeman, W. J. (2000). The psychometric properties of the Learning Style Inventory and the Learning Style Questionnaire: Two normative measures of learning styles. *South African Journal of Psychology*, 30 (2), 44–52.
172. Plečkaitis, A., Žvirblis, E. (2013). *IRT specialistų Lietuvoje pasiūlos-paklausos prognozė 2014–2016 m.* [žiūrėta 2012 m. gruodžio 10 d.]. Prieiga per internetą: <http://www.infobalt.lt/lt/activities/initiatives/100>.
173. Pollack, S., Schertz, Z. (2003). Supporting project development in CS – the effect on intrinsic and extrinsic motivation. In *Proceedings of the Eleventh International PEG Conference.* St Petersburg, Russia.
174. Preidys, S., Sakalauskas, L. (2010). Analysis of Students’ Study Activities in Virtual Learning Environments Using Data Mining Methods. *Technological and Economic Development of Economy*, 16, 94–108.

-
175. Preidys, S., Žilinskienė, I. (2012). An e-learning personalization model from learning activities perspective. In *Electronic learning, information and communication: theory and practice* (1) (pp. 111–132). Vilnius: Vilniaus universiteto Elektroninių studijų ir egzaminavimo centras.
 176. Prensky, M. (2013). *Digital natives, digital immigrants*. Cross Currents: Cultures, Communities, Technologies.
 177. Pukelis, K. (2009). Gebėjimas, kompetencija, mokymosi / studijų rezultatas, kvalifikacija ir kompetentingumas: teorinė dimensija. *Aukštojo mokslo kokybė*, 6, 12–34.
 178. Pukelis, K., Pileičikienė, N. (2010). Bendrųjų mokėjimų ugdymo gerinimas aukštųjų mokyklų studijų programose: absolventų požiūris. *Aukštojo mokslo kokybė*, 7, 108–131. [žiūrėta 2013 m. gruodžio 11 d.]. Prieiga per internetą: <http://files.eric.ed.gov/fulltext/EJ900261.pdf>.
 179. Pukėnas, K. (2009). *Kokybinių duomenų analizė SPSS programa*. Kaunas: Lietuvos kūno kultūros akademija.
 180. Railienė, A. (2010). *Profesijos patarėjo kompetencijų ugdymas rengiant socialinius pedagogus: daktaro disertacija*. Vilnius: Vilniaus pedagoginio universiteto leidykla.
 181. Rajeckas, V. (1999). *Mokymo organizavimas*. Kaunas: Šviesa.
 182. Ramsden, P. (2000). *Kaip mokyti aukštojoje mokykloje*. Vilnius: Aidai.
 183. Reason, R. W., Bradburry, H. (2007). *The SAGE handbook of Action research*. Thousand Oaks, Sage Publications.
 184. Riding, R., Cheema, I. (1991). Cognitive styles: An overview and integration. *Educational Psychology*, 11, 193–215.
 185. Robins, A., Rountree, J., Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13, 2, 137–172.
 186. Rupšienė, L. (2000). *Nenoras mokytis – socialinis pedagoginis reiškinys*. Klaipėda: Klaipėdos universiteto leidykla.
 187. Russell, G. (2004). *Herrmann Brain Dominance Instrument*. The Ned Herrmann Group, Inc. 23.
 188. Sadler-Smith, E. (2001). The relationship between learning style and cognitive style. *Personality and Individual Differences*, 30 (4), 609–616.
 189. Sajaniemi, J., Kuittinen, M. (2005). An experiment on using roles of variables in teaching introductory programming. *Computer Science Education*, 15 (1), 59–82.
 190. Schmeck, R. R. E. (1988). *Learning strategies and learning styles*. Plenum Press.
 191. Sharp, J. H., Schultz, L. A. (2013). An Exploratory Study of the use of Video as an Instructional Tool in an Introductory C# Programming Course. *Information Systems Education Journal*, 11 (6) 33–39. [žiūrėta 2014 m. kovo 10 d.]. Prieiga per internetą: <http://isedj.org/2013-11/> ISSN: 1545-679X.
 192. *Should we be using learning styles? What research has to say to practice*. [žiūrėta 2011 m. gruodžio 11 d.]. Prieiga per internetą: <http://www.lsda.org.uk/files/PDF/1540.pdf>.
 193. Sims, R. R., Veres, J. G., Shake, L. G. (1989). An exploratory examination of the convergence between the Learning Styles Questionnaire and the Learning Style Inventory II. *Educational and Psychological Measurement*, 49 (1), 227–233.
 194. Sorva, J. (2010). Reflections on Threshold Concepts in Computer Programming and Beyond. In *Proceedings of the Koli Calling International Conference on Computing Education Research, Koli Calling' 10* (pp. 21–30). ACM.
 195. Sorva, J. (2012). *Visual Program Simulation in Introductory Programming Education*. Ph. D. dissertation, Aalto University.
-

196. Stash, N. V., Cristea, A. I., De Bra, P. M. (2004). Authoring of learning styles in adaptive hypermedia: problems and solutions. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters* (pp. 114–123). ACM.
197. Sung, K., Hillyard, C., Angotti R. L., Panitz, M. W., Goldstein, D. S., Nordlinger, J. (2010). Game-Themed Programming Assignment Modules: A Pathway for Gradual Integration of Gaming Context Into Existing Introductory Programming Courses. *IEEE Transactions on Education, PP*, Issue 99, 1–12.
198. Swailes, S., Senior, B. (1999). The dimensionality of Honey and Mumford's learning styles questionnaire. *International Journal of Selection and Assessment*, 7 (1), 1–11.
199. Šiaučiukėnienė, L. (2013). *Kauno suaugusiųjų mokymo centras*. [žiūrėta 2014 m. sausio 7 d.]. Prieiga per internetą: <http://ksmc.lt/77-mokymo-individualizavimo-ir-diferencijavimo-teorines-izvalgos>.
200. Šiaučiukėnienė, L., Visockienė, O., Talijūnienė, P. (2006). *Šiuolaikinės didaktikos pagrindai*. Kaunas: Technologija.
201. Tepper, B. J., Tetrault, L. A., Braun, C. K., Romero, J. E. (1993). Discriminant and convergent validity of the problem solving style questionnaire. *Educational and Psychological Measurement*, 53 (2), 437–444.
202. Teresevičienė, M., Bulajeva, T., Čepienė, A., Lepaitė, D., Zuzevičiūtė, V. (2011). *Studijų programų atnaujinimas: kompetencijų plėtotės ir studijų siekinių vertinimo metodika*. Vilnius: Vilniaus universiteto leidykla.
203. Teresevičienė, M., Gedvilienė, G. (1999). *Mokymasis bendradarbiaujant*. Vilnius, Garnelis.
204. Teresevičienė, M., Gedvilienė, G. (2003). *Mokymasis grupėse ir asmenybės kaita*. Kaunas: Garnelis.
205. Teresevičienė, M., Gedvilienė, G., Zuzevičiūtė, V. (2006). *Andragogika*. Kaunas: VDU leidykla.
206. Teresevičienė, M., Oldroyd, D., Gedvilienė, G. (2004). *Suaugusiųjų mokymasis. Andragogikos didaktikos pagrindai*. Kaunas: VDU leidykla.
207. *The European Qualification Framework for Lifelong Learning (EQF)*. (2008). European Commission, Education and Culture, Luxembourg.
208. Theodorou, C., Kordaki, M. (2010). Super Mario: a Collaborative Game for The Learning of variables in Programming. *International Journal of Academic Research*, 2, Issue 4, 111–118.
209. Thompson, E., Luxton-Reilly, A., Whalley, J. L., Hu, M., Robbins, P. (2008). Bloom's taxonomy for CS assessment. In S. Hamilton and M. Hamilton (Eds.). *Proceedings of the tenth conference on Australasian computing education*, 78 (ACE '08), 155–161. Australian Computer Society, Inc., Darlinghurst, Australia.
210. Thomas, L., Ratcliffe, M., Woodbury, J., Jarman, E. (2002). Learning styles and performance in the introductory programming sequence. *ACM SIGCSE Bulletin*, 34, 1, 33–37. ACM.
211. Tillmann, N., de Halleux, J., Tao, X., Gulwani, S., Bishop, J. (2013). Teaching and learning programming and software engineering via interactive gaming. In *Software Engineering (ICSE), 2013 35th International Conference* (pp. 1117, 1126).
212. *Tyrimo „Mokytojų didaktinės kompetencijos atitiktis šiuolaikiniams švietimo reikalavimams“ ataskaita*. (2005). Kaunas: KTU.
213. Urbonienė, J. (2012). Programavimo mokymo aplinkų efektyvumo mokymo(si) atžvilgiu analizė. *Studijos šiuolaikinėje visuomenėje*, 3 (1), 243–254.
214. Urbonienė, J. (2013). Adaptyvios programavimo mokymosi sistemos modelis. *Informacijos mokslai*, 66, 108–122.

-
215. Urbonienė, J. (2014). *Adaptyviųjų programavimo mokymo priemonių projektavimas: daktaro disertacija*. Vilnius: Vilniaus universiteto leidykla.
 216. Van-Roy, P., Haridi, S. (2002). *Teaching programming broadly and deeply: The kernel language approach, Informatics Curricula and Teaching Methods*, 53–62.
 217. Van-Roy, P., Haridi, S. (2004). *Concepts, Techniques and Models of Computer Programming*. MIT Press.
 218. Van Zwanenberg, N., Wilkinson, L. J., Anderson, A. (2000). Felder and Silverman's Index of Learning Styles and Honey and Mumford's Learning Styles Questionnaire: how do they compare and do they predict academic performance? *Educational Psychology*, 20 (3), 365–380.
 219. Vargas, M. V., Jimenez, J. M., Santamaria, M. L. S., Navarro, J. L. A., Aviles, G. F. (2011). Cooperative Learning In *Virtual Environments: The Jigsaw Method In Statistical Courses. Journal of International Education Research*.
 220. Vidžiūnas, A., Mickus, A. (2007). Integruoto programavimo koncepcijų mokymo C++ kalba principai. *Informacijos mokslai*, 42–43, 195–200.
 221. Vitkutė-Adžgauskienė, D., Vidžiūnas, A. (2012). Problems in Choosing Tools and Methods for Teaching Programming. *Informatics in education*, 11 (2), 271–282.
 222. Vujošević-Janičić, M., Tošić, D. (2008). The role of programming paradigms in the first programming courses. *The Teaching of Mathematics*, XI, 2, 63–8.
 223. Waleed, F., Sanwar, A., Debzani, D. (2013). Does language choice influence the effectiveness of online introductory programming courses? In *Proceedings of the 14th annual ACM SIGITE conference on Information technology education (SIGITE '13)*. ACM, New York, NY, USA.
 224. Windschitl, M. (2002). Framing constructivism in practice as the negotiation of dilemmas: An analysis of the conceptual, pedagogical, cultural, and political challenges facing teachers. *Review of educational research*, 72 (2), 131–175. [žiūrėta 2015 m. sausio 18 d.]. Prieiga per internetą: <http://rer.sagepub.com/content/72/2/131.short>.
 225. Winslow, L. E. (1996). Programming pedagogy, a psychological overview. *ACMSIGCSE Bull.*, 28, Issue 3, 17–22.
 226. Wulf, T. (2005). Constructivist approaches for teaching computer programming. In *SIGITE'05* (pp. 245–248). October 20–22, Newark, New Jersey, USA.
 227. Zacharis, N. Z. (2011). Measuring the Effects of Virtual Pair Programming in an Introductory Programming Java Course. *IEEE Transactions on Education*, 54, 1, 168–170.
 228. Zuzevičiūtė, V., Teresevičienė, M. (2007). *Universitetinės studijos mokymosi visą gyvenimą perspektyvoje*. Kaunas: VDU leidykla.
 229. Паниотто, В. И., Максименко, В. С. (2003). *Количественные методы в социологических исследованиях*. Киев: Наукова Думка.

PRIEDAI

- 1 priedas.** Lentelių sąrašas
- 2 priedas.** Paveikslų sąrašas
- 3 priedas.** Anketa studentui
- 4 priedas.** Įvadinio programavimo kursą dėstančių dėstytojų interviu klausimų sąrašas
- 5 priedas.** Veiklos tyrimo dalyvių interviu klausimų sąrašas
- 6 priedas.** Įvadinio programavimo kurso išplėstinis planas
- 7 priedas.** Stebėjimo protokolo forma ir užpildyto protokolo pavyzdys
- 8 priedas.** Naudotos vertinimo ir įsivertinimo formos
- 9 priedas.** Programavimui mokytis taikomų mokymosi metodų sąrašas
- 10 priedas.** Mokymosi stilių ir mokymosi metodų dermės modelis
- 11 priedas.** Mokymosi stilių nustatymo klausimynas

1 PRIEDAS

Lentelių sąrašas

1 lentelė	Mokymosi stiliaus apibrėžimai.....	17
2 lentelė	F. Coffieldo mokymosi stilių teorijų taksonomija (Coffield ir kt., 2004)	19
3 lentelė	Ugdymo praktikoje naudojamos mokymosi stilių tipologijos.....	20
4 lentelė	Mokymosi ciklo etapų ir mokymosi metodų sąsajos	42
5 lentelė	Atnaujintos Bloomo taksonomijos lygmenys ir juos atitinkančios mokymosi veiklos.....	43
6 lentelė	Atnaujintos Bloomo taksonomijos taikymas mokantis programavimo (Thompson ir kt., 2008).....	45
7 lentelė	Mokymosi stiliams būdingos ypatybės (Honey, Mumford, 2000).....	53
8 lentelė	Mokymosi stilių ir mokymosi veiklų sąsajos (Honey, Mumford, 2000)	54
9 lentelė	Mokymosi stilių ir mokymosi metodų klasifikacija.....	55
10 lentelė	Mokymosi stilių ir mokymosi metodų dermės modelis.....	56
11 lentelė	Studijų programų pasiskirstymas pagal studijų kryptis.....	60
12 lentelė	Priimtųjų į IK kryptių programas 2011–2013 metais skaičiai.....	62
13 lentelė	Demografinės informantų charakteristikos.....	64
14 lentelė	P. Honey ir A. Mumfordo pateiktas mokymosi stiliaus raiškos stiprumas (Honey, Mumford, 2000)	68
15 lentelė	Tiriamųjų, dalyvavusių veiklos tyrime (N = 39), pasiskirstymas pagal jų mokymosi stilius.....	69
16 lentelė	Stebėtų informantų mokymosi stilių išreikštumas ir priskyrimas mokymosi stilių grupėms.....	70
17 lentelė	Atskirų P. Honey ir A. Mumfordo klausimyno dalių vidinis suderinamumas (N = 74).....	72
18 lentelė	Programavimo mokymosi paradigmos Lietuvos aukštosiose mokyklose.....	77
19 lentelė	Programavimo kalbos pasirinkimas Lietuvos aukštosiose mokyklose.....	77
20 lentelė	Pirmosios programavimo kalbos pasirinkimas pagal studentų (2013 m.) skaičių	78

21 lentelė	ĮPK tikslų analizė.....	80
22 lentelė	Kurso „Įvadas į programavimą“ (Kreitono universitetas) ir analizuotų ĮPK aprašų tematikos palyginimas	81
23 lentelė	ĮPK studijų rezultatai	82
24 lentelė	ĮPK dalyko studijų metu taikytų mokymosi metodų dažnis pagal aukštąsias mokyklas.....	96
25 lentelė	Mokymosi metodų vertinimai.....	106
26 lentelė	Mokymosi metodų vertinimai pagal mokymosi stilius MSMMD modelyje.....	109
27 lentelė	Mokymosi motyvacijos įvertinimas	112
28 lentelė	Algoritmų ir programų vizualizavimo metodo vertinimų kategorizavimas	114
29 lentelė	Darbo grupėse bendradarbiaujant vertinimų kategorizavimas.....	116
30 lentelė	Programavimo poromis metodo vertinimų kategorizavimas	118
31 lentelė	Diskusijų metodo vertinimų kategorizavimas	120
32 lentelė	Tradicinių paskaitų metodo vertinimų kategorizavimas	122
33 lentelė	Demonstravimo metodo vertinimų kategorizavimas.....	123
34 lentelė	Programos teksto skaitymo ir veikimo sekimo metodo vertinimų kategorizavimas	124
35 lentelė	Individualių programavimo užduočių vertinimų kategorizavimas	126
36 lentelė	Pristatymų metodo vertinimų kategorizavimas	128
37 lentelė	Savikontrolės testų vertinimų kategorizavimas.....	130
38 lentelė	Vaizdo pamokų vertinimo kategorizavimas	131
39 lentelė	Varžybų metodo vertinimų kategorizavimas.....	132

2 PRIEDAS

Paveikslų sąrašas

1 pav.	Programavimo paradigms klasifikacija.....	26
2 pav.	Galimos įvadinio programavimo kurso dedamosios dalys pagal CSC2013.....	31
3 pav.	Programavimo proceso (programavimo užduoties sprendimo) etapai	34
4 pav.	Pradedančiųjų programavimo gebėjimai, lemiantys sklandžią programavimo proceso eigą.....	36
5 pav.	J. T. Mayeso ir C. J. Fowlero (1999) mokymosi ciklas	40
6 pav.	Mokslinio tyrimo struktūrinė schema.....	48
7 pav.	MSMMD modeliavimo procesas	52
8 pav.	Mokymosi stilių tipologija (Honey, Mumford, 2000)	54
9 pav.	Programavimo gebėjimų ugdymo turinio tyrimo struktūrinė schema....	59
10 pav.	Anketinio tyrimo populiacija ir imtis	63
11 pav.	Atlikto veiklos tyrimo schema	66
12 pav.	Studijų metodai ĮPK aprašuose.....	85
13 pav.	Vertinimo metodai ĮPK aprašuose.....	86
14 pav.	Respondentų programavimo patirtis iki studijų aukštosiose mokyklose ..	88
15 pav.	Respondentų programavimo patirtis iki studijų pagal studijų kryptis.....	89
16 pav.	Respondentų turima programavimo patirtis pagal programavimo kalbas iki studijų.....	90
17 pav.	Studentų nuomonė apie programavimo mokymosi sudėtingumą aukštosiose mokyklose.....	91
18 pav.	Respondentų nurodomi programavimo mokymosi sunkumai.....	92
19 pav.	Respondentų nurodomi programavimo mokymosi sunkumai pagal studijų kryptis.....	92
20 pav.	Programavimo mokymosi sunkumų ir patirties ryšys.....	93
21 pav.	Studijuojant ĮPK taikyti studijų metodai.....	94
22 pav.	Respondentų pageidaujami mokymosi būdai	98
23 pav.	Respondentų pageidaujami mokymosi būdai kolegijose ir universitetuose....	98
24 pav.	Studentų požiūris į taikytus mokymosi metodus.....	107
25 pav.	Studentų požiūris pagal mokymosi stilių tipus.....	108
26 pav.	Prieštaraujančių MSMMD modeliui vertinimų pasiskirstymas pagal mokymosi stilių tipus.....	111

3 PRIEDAS

Anketa studentui

Atliekamas tyrimas apie įvadinio programavimo (tai pirmasis su programavimu susijęs dalykas, kurį studijavote aukštojoje mokykloje) mokymąsi ir studentų požiūrį į šio dalyko studijas. Tai padės tobulinti pradedančiųjų programavimo mokymąsi. Mums svarbi Jūsų nuomonė, todėl prašome nuoširdžiai atsakyti į pateikiamus klausimus. Anketa yra anoniminė. Atsakymai bus naudojami apibendrinti bei skelbiami tik apdoroti. Iš anksto dėkojame už nuoširdumą ir sugaištą laiką.

Lietuvos edukologijos universitetas
Informatikos katedra

1. Tavo lytis:

- Vyras
- Moteris

2. Tavo amžius (įrašyk)

3. Tavo gyvenamoji vieta (įrašyk)

4. Kokios aukštosios mokyklos studentas esi? (įrašyk)

5. Įrašyk studijų programos, kurią studijuoji, pavadinimą.

(įrašyk)

6. Ar Tau patinka programuoti?

- Ne
- Nežinau
- Taip

7. Ar anksčiau (iki studijų) teko susidurti su programavimu?

- Ne
- Taip

8. Kur anksčiau teko susidurti su programavimu? (jeigu neteko, klausimą praleisk) (daug galimų variantų):

- Mokykloje
- Neakivaizdinėje programuotojų mokykloje (pvz., JPM ar kitoje)
- Kitoje mokykloje (pvz., privačioje kompiuterių mokykloje ar pan.)
- Programuoju savarankiškai
- Kita (įrašyk).....

9. Kokiomis programavimo kalbomis teko programuoti anksčiau (iki studijų aukštojoje mokykloje)? (jeigu neteko, klausimą praleisk) (daug galimų variantų):

- C
- C++
- Visual Basic
- Java
- Python
- Pascal
- Php
- kita (įrašyk).....

10. Kelintais metais studijavai pirmąjį programavimo dalyką aukštojoje mokykloje?

- 2011 m.
- 2012 m.
- 2013 m.
- Kita (įrašyk).....

11. Kokią programavimo kalbą studijavai kaip pirmąją programavimo kalbą aukštojoje mokykloje?

- C
- C++
- Visual Basic
- Java
- Python
- Pascal
- Php
- Kita (įrašyk).....

12. Ar įvadinis programavimo kursas padėjo pagerinti programavimo įgūdžius?

- Ne
- Taip
- Kita (įrašyk).....

13. Kokius programavimo mokymo ir mokymosi metodus taikai studijų metu ir kaip dažnai?

	Niekada	Vieną kartą per savaitę arba rečiau	Daugiau nei vieną kartą per savaitę
Klausiau dėstytojo skaitomų paskaitų	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sprendžiau uždavinius individualiai	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Paskaitų metu dirbau mažose komandose	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Paskaitų metu dirbau poroje su kitais studentais	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Braižiau programos veikimo blokines schemas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Atlikau projektinę užduotį	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Žiūrėjau <i>YouTube</i> vaizdo pamokas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Skaičiau forumus	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Naudojau specialios paskirties mokomąsias programas ir aplinkas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mokiausi papildomai samdydamas korepetitorių	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Skaičiau knygas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Studijavau internetiniuose mokymosi kursuose	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sprendžiau testus	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Konsultavausi su dėstytoju	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kita (įrašykite).....			

14. Ar naudojai programavimui mokytis skirtas kompiuterines mokomąsias priemones (pvz.: *trakla*, *rurple*, *code.org* ar kt.). Jei taip, išvardyk. (Jeigu nenaudojai, klausimą praleisk.)

(įrašyk)

15. Ar, Tavo manymu, sunku yra išmokti programuoti?

- Ne
- Nežinau
- Taip

16. Kodėl, Tavo manymu, iškyla sunkumų mokantis programavimo? (daug galimų variantų)

- Besimokantieji nepakankamai daug dirba savarankiškai
- Besimokantiems sunku gauti ne patį rezultatą, t. y. parašyti programą, bet pereiti patį kūrimo procesą

- Besimokantieji neteisingai transformuoja natūralios kalbos semantiką į programavimo kalbą
- Programavimas reikalauja aukšto abstrakcijos lygio
- Programavimo kalbų sintaksė yra gana sudėtinga
- Besimokantieji neturi motyvacijos
- Kita (įrašyk).....

17. Koks programavimo mokymo(si) būdas Tau priimtinausias?

- Kai dėstytojas aiškina grupei studentų
- Kai dėstytojas konsultuoja individualiai
- Neakivaizdinis mokymosi būdas
- Nuotolinis mokymosi būdas
- Mokymasis naudojant interaktyvias ar adaptyvias aplinkas
- Mokymasis naudojant įvairias technologijas
- Kita (įrašyk).....

4 PRIEDAS

ĮPK dėstytojų interviu klausimų sąrašas

- Kokia Jūsų kaip programavimo dėstytojo patirtis?
- Kokią programavimo kalbą dėstote?
- Ar tai pirmasis programavimo dalykas?
- Koks studentų grupės dydis?
- Trumpai apibūdinkite, kaip dėstote programavimo pagrindus semestro metu.
- Ar bandote individualizuoti mokymą, atsižvelgti į studentų gebėjimus, mokymosi stilių ir panašiai?
- Ar naudojate kokias nors technologijas (pvz.: *moodle*, *trakla* ir kt.)?
- Kaip Jūs manote, kam turi būti teikiamas prioritetas mokant pradedančiuosius programavimo?
- Kas, Jūsų manymu, studentams sunkiausia?
- Kaip vertinate mokymo, atsižvelgiant į mokymosi stilių (paaiškinu), galimybę nuolatinėse studijose?
- Jūsų pastabos ir komentarai.

5 PRIEDAS

Veiklos tyrimo dalyvių interviu klausimų sąrašas

- Kokia buvo Tavo programavimo patirtis įstojus mokytis į aukštąją mokyklą?
- Kas yra programavimas Tau asmeniškai?
- Ar sunku buvo mokytis programuoti ir kaip manai kodėl? Pabandyk įvardyti temas ar situacijas, kuriose patyrei sunkumų? Kaip juos sprendei?
- Įvertink programavimo užduoties etapus (užduoties analizė, algoritmo sudarymas, algoritmo užrašymas programavimo kalba, testavimas ir derinimas), kas Tau sunkiausia?
- Įvertink programavimo mokymosi metodus (tradicinė paskaita, aiškinimas ir demonstracijos, vaizdo pamokos, savikontrolės testas, diskusijos, pavyzdžių analizė, programos teksto skaitymas, programos veikimo sekimas, komandinis darbas mažose grupėse, programavimas porose, darbo pristatymas, programos veikimo vizualizavimas, schemas, struktūrogramos, individuali programavimo užduotis, varžybos, savarankiškas darbas ir kt.), kuriuos taikei paskaitų metu ir dirbdamas savarankiškai, pakomentuok, kodėl patiko / nepatiko juos taikyti.
- Tavo pastabos ir komentarai.

6 PRIEDAS

Išplėstinis dalyko „Programavimo pagrindai“ planas

Dalyko tikslas

- Ugdyti gebėjimus analizuoti užduotis ir kurti programas įvairiems uždaviniams spręsti, parenkant, pritaikant, sudarant uždavinių realizavimo algoritmus, užrašant juos C++ programavimo kalba, redaguojant sintaksės klaidas, vykdančiant ir testuojant programas, organizuojant aiškia programos vykdymo sąsają.

Vertinimas

- Įgytos žinios ir gebėjimai vertinami, kai pasiekiami studijų rezultatai pagal BLOOMO taksonomijos lygmenis. Dešimtbalėje vertinimo sistemoje vertinami testai, savarankiški, praktiniai, grupėse ir porose atlikti darbai.

Temos	Valandos (nuo–iki)	Studijų rezultatai	Dėstytojo veiklos	Studentų veiklos
1. Įvadas.	2		Pristato dalyko mokymosi tikslą, numatytą ugdymo turinį, vertinimo sistemą. Paaškina mokymosi stiliaus vaidmenį mokantis. Išsiaiškina studentų pasirengimą dalyko studijoms.	Užpildo mokymosi stiliaus įsivertinimo anketą.
2. Programavimo kalbos ir aplinkos. Algoritmų sudarymas. Pagrindiniai uždavinio sprendimo etapai. Programos struktūra, vykdymas pasirinktoje programavimo aplinkoje (Dev C++, CodeBlocks). Konstantos ir kintamieji. Paprastieji duomenų tipai, operatoriai, jų prasmė ir naudojimo būdai, operatorių pirmenybė ir kintamojo tipo keitimas, įvedimo / išvedimo srutai, veiksmai, standartinės aritmetinės funkcijos. Tiesiniai ir šakotieji algoritmai. Sąlyginiai sakiniai <i>if</i> , <i>if else</i> , daugiavariantės alternatyvos operatorius <i>switch</i> .	4	Susisteminta turimas žinias, analizuojant užduotį parenka tinkamus duomenų tipus, numato užduoties sprendimo metodus ir sudaro skaičiavimų algoritmus. Geba naudotis programavimo aplinka.	Vadovauja diskusijai, padeda susisteminti pagrindines mokyklinio kurso žinias. Paaškina dažniausiai daromas sintaksės klaidas.	Minčių žemėlapis; diskusija. Jeigu studentai pažįstami – gali būti organizuojamas darbas grupėse. Pratybos. Surenka duotą programos tekstą, kompiliuoja, vykdo. Išsiaiškina kompiliatoriaus pranešimus. Kuria tiesinius ir šakotuosius algoritmus.

Temos	Valandos (nuo–iki)	Studijų rezultatai	Dėstytojo veiklos	Studentų veiklos
3. Cikliniai algoritmai. C++ ciklinės struktūros. Tipiniai algoritmai (kaupiamoji suma, sandauga, kiekis, didžiausia, mažiausia reikšmės, duomenų sekos įvedimas) ir jų realizavimas C++ programavimo kalba. Programų analizės priemonės. Programavimo kultūra, stilius.	5–6	Analizuoja užduotį, parengia užduoties realizavimo schemą, taiko tipinius ciklinius algoritmus.	Teorinė dalis pateikiama glaustai pristatant užduotį, tam skiriamas daugiau kaip 5–10 min. Konsultuoja analizuojant užduotį, netiesioginiais klausimais bei komentais nukreipia ir / ar pataria parenkant sprendimo būdus, skatina savarankiškai priimti sprendimus.	Užduočių analizė, sprendimo schemos parengimas. Pratybos. Algoritmų (kaupiamosios sumos, sandaugos, tam tikrą požymį atitinkančio reikšmių kiekio) taikymai. Darbas porose.
4. Tekstiniai failai. Atsitiktinių skaičių parinkimas.	3–4	Pradiniamis duomenimis ir programos veikimo rezultatais saugoti naudoja tekstinis failus.	Teorinė dalis apie tekstinis failus pateikiama glaustai pristatant užduotį, tam skiriamas daugiau kaip 5–10 min. Atsitiktinių skaičių generavimui išsiaiškinti nurodo galimus informacijos paieškos reikšminius žodžius. Konsultuoja sprendžiant užduotis.	Pratybos. Kuria duomenų failus (naudoja atsitiktinius skaičius) ir juos analizuoja. Atlieka informacijos paiešką apie atsitiktinių skaičių generavimą internete. Analizuoja surastus programinius sprendimus. Įgyvendinami žaidybiniai sprendimai (pvz., atspėk skaičių).
5. Programos skaidymas į dalis: funkcijos. Funkcijos struktūra. Globalieji, lokalieji ir statiniai kintamieji. Funkcijų parametrai, makrokomandos, iš anksto nusakomi funkcijos parametrai. Rekursija.	7–8	Analizuoja užduotį, geba ją skaidyti į dalis, nustatyti ryšius tarp jų, parengti sprendimo schemą. Geba išskirti dalis užrašyti funkcijomis ir jas pritaikyti programose.	Teorinė dalis pateikiama glaustai pristatant užduotį, tam skiriamas daugiau kaip 10–15 min. Organizuoja ir vadovauja darbui grupėse (ekspertų metodas).	Pratybos. Grupėse sprendžia mažas užduoties dalis, vėliau – didesnes užduotis, pristato kitiems, vertina ir įsivertina atliktą darbą.
6. Masyvas. Masyvo formavimas, didžiausios (mažiausios) reikšmių paieška, rikiavimas, veiksmai su masyvo elementais.	5–6	Geba suformuoti masyvą ir atlikti veiksmus su jo elementais.	Teorinė dalis pateikiama glaustai pristatant užduotį, tam skiriamas daugiau kaip 10–15 min. Konsultuoja sprendžiant užduotis.	Pratybos. Algoritmų (reikšmių, atitinkančių tam tikrą požymį, paieška, rikiavimas, elementų įterpimas ir šalinimas) taikymai. Darbas porose.

Temos	Valandos (nuo-iki)	Studijų rezultatai	Dėstytojo veiklos	Studentų veiklos
7. Tekstiniai failai. Teksto analizė ir tvarymas. Simboliai, duomenų tipas <i>char</i> , <i>char</i> simbolių eilutės ypatybės, kai kurios svarbesnės simbolių eilučių apdoravimo funkcijos. <i>String</i> simbolių eilutės aprašymas, reikšmių jai suteikimas. Veiksmai su <i>string</i> simbolių eilutėmis. <i>String</i> simbolių eilučių masyvai.	9–10	Geba atlikti veiksmus su simboliais ir simbolių eilutėmis. Geba atlikti reikiamų simbolių eilučių apdoravimo funkcijų aprašą pateiška.	Teorinė dalis pateikiama glaustai pristatant užduotį, tam skiriamas daugiau kaip 5–10 min. Pristatoma simbolių eilučių apdoravimo funkcijų žinyną. Konsultuoja sprendžiant užduotis.	Pratybos. Tekstinės informacijos skaitymo (rašymo) iš (į) failo pagal duotą užduotį algoritmas ir jo įgyvendinimas programavimo kalba. Virtualiojoje mokymosi aplinkoje kuria naudotų funkcijų žinyną (vikį). Teksto analizė ir keitimas pagal duotą užduotį. Darbas grupėse arba individualiai. Savarankiškai arba grupėse atlikto darbo pristatymas ir / ar gynimas.
8. Dvimaciai ir daugiamačiai masyvai. Elementų paieška, elementų ar jų grupių šalinimas, įterpimas, sukeitimas, rikiavimas. Veiksmai su dvimacio masyvo dalimis. Kvadratinė matrica. Elementų, esančių skirtingose įstrižainėse, nagrinėjimas. Elementai tarp įstrižainių, įstrižainių susikirtimas.	5–6	Geba suformuoti dvimatių masyvą ir atlikti veiksmus su jo elementais.	Teorinė dalis pateikiama glaustai pristatant užduotį, tam skiriamas daugiau kaip 10–15 min. Konsultuoja sprendžiant užduotis	Pratybos. Algoritmų (reikšmių, atitinkančių tam tikrą požymį, paieška, rikiavimas, elementų įterpimas ir šalinimas) taikymai. Projektinis darbas arba porose, arba individualiai.
9. Rodyklės ir jų naudojimo ypatumai.	4–8	Supranta rodyklių paskirtį ir geba jomis naudotis.	Teorinė dalis pateikiama glaustai pristatant užduotį, tam skiriamas daugiau kaip 5–10 min. Konsultuoja sprendžiant užduotis.	Pratybos. Užrašo anksčiau sukurtas programos, naudojamas rodyklėmis. Darbas porose arba individualiai.

Temos	Valandos (nuo–iki)	Studijų rezultatai	Dėstytojo veiklos	Studentų veiklos
10. Struktūros.	7–8	Geba atlikti veiksmus su struktūromis ir masyvais su struktūros tipo elementais.	Teorinė dalis pateikiama glaustai pristatant užduotį, tam skiriamas daugiau kaip 5–10 min. Vadovaujama bendrai sprendžiant uždavinį. Konsultuoja sprendžiant užduotis.	Pratybos. Remiantis pavyzdžiu kuria savo struktūrą, organizuoja duomenų įvedimo procesą, atlieka pateiktą struktūrų masyve. Projektinis darbas porose arba individualiai.
11. Įvadas į objektinį programavimą. Objektinio programavimo samprata. Klasės ir objektai. Konstruktorius, destruktorius, klasių, metodų modifikatoriai.	2–4	Supras objektinio programavimo principus.	Teorinė dalis pateikiama glaustai pristatant užduotį, tam skiriamas daugiau kaip 5–10 min. Konsultuoja sprendžiant užduotis.	Pratybos. Remiantis pavyzdžiu kuria savo klasę ir naudoja programoje.
	62			

7 PRIEDAS

Stebėjimo protokolas

[data] protokolas NR.

Dalyvių sąrašas:

Paskaitos tikslas – ...

Taikomi mokymosi metodai: ...

Paskaitos tema: ...

Laikas (nuo–iki)	Dėstytojo veiklos	Studentų veiklos	Stebėjimo pastabos

Tyrėjo pastabos:....

Užpildyto protokolo pavyzdys

2012-11-06 protokolas NR. 11

Dalyvių sąrašas:

P1 – mąstytojas, teoretikas, mokėsi mokykloje *Pascal*;

P2 – mąstytojas (aktyvistas), neturi programavimo patirties;

P3 – aktyvistas, mokėsi mokykloje *Pascal*;

P4 – pragmatikas, mokėsi mokykloje *Pascal*;

P5 – mąstytojas, šiek tiek žino *Pascal*;

P6 – pragmatikas, šiek tiek žino *Pascal*;

P7 – neišreikštas, mokėsi mokykloje *Pascal*.

Ir trys studentai S1, S2, S3, nedalyvaujantys tyrime.

Paskaitos tikslas – ugdyti ciklinių algoritmų užrašymo ir aiškinimo gebėjimus, išsiaiškinti atsitiktinių skaičių parinkimą ir taikyti tai sprendžiant užduotis.

Taikomi mokymosi metodai: durstinys, programavimas porose, pristatymai.

Paskaitos tema: Ciklai, duomenų srauto įvedimas ir apdorojimas. Atsitiktinių skaičių generavimas.

Laikas (nuo iki)	Dėstytojo veiklos	Studentų veiklos	Pastabos
5 min.	Pristato paskaitos tikslą, paaiškina durstinio metodą. Suskirsto studentus į grupes po 5: 1. Programuotojai P3, P4, P7, S2, S3; 2. Teoretikai-mąstytojai P1, P2, P5, P6, S1.	Klausosi, užduoda klausimus.	Į abi grupes pakliuvo po pragmatiką.
20 min.	Organizuoja ir prižiūri, kai reikia, konsultuoja.	Programuotojai rašo programą: 1. Ciklas, kuris nutraukiamas, kai vartotojas pageidauja, pvz., įveda nulį; 2. Cikle įvedamas duomenų srautas, ciklo pabaiga, kai įvedama x duomenų reikšmių, atitinkančių pasirinktą sąlygą (pvz., lyginiai skaičiai; teigiami skaičiai ir pan.). Teoretikai ieško informacijos internete apie tai, kaip programoje sugeneruoti bet kokią sveiką skaičių. Parašo trumpą programėlę (arba suranda ir pritaiko sau), kuri generuoja triženklus natūraliuosius skaičius.	Programuotojų grupė nedarni (galbūt per daug nuomonių), individualiai dirba S3, P7, P3. S2 ir P4 dirba kartu. P7 susigalvoja užduočių sau pats; improvizuoja daug, bet ne iki galo užrašo savo sugalvotų užduočių sprendimus. P1, P2, P6 dirba prie vieno kompiuterio, lyderio aiškina irgi nėra, bet tariasi ir bando įtraukti dėstytoją. P5 ir S1 daugiau dirba atskirai.
20 min.	Išardomos grupės, o studentai suporuojami: 1 pora: P6 + P7; 2 pora: P1 + P4; 3 pora: S1 + S3; 4 pora: P2 + S2; 5 pora: P5 + P3.	Poros gauna sujungtas užduotis, kurias atliekant reikia pritaikyti abiejų patirtį.	
15 min.	PERTRAUKA		
15 min.		Poros pristato atliktas užduotis, pakomentuoja, ką ir kaip darė.	4 pora užduoties neatliko, nedidelių klaidų padarė 1 ir 5 poros. Savarankiškiau-siai dirbo 2 pora.
5 min.	Aptaria atliktas užduotis.	Aptaria atliktas užduotis.	
25 min.	Skiria individualias panašaus pobūdžio užduotis, konsultuoja individualiai.	Atlieka individualias užduotis. Rezultatai pateikiami VMA moodle, studentai pildo anketą.	

Lyginant su įprastu paskaitos vedimo būdu, užduočių išspręsta mažiau, tačiau patiko tai, kaip jie įsitraukė ir dirbo visos paskaitos metu. Paskaitos pradžioje sudarytos didelės grupės nebuvo itin sėkmingos ir darbas pasidarė daugiau individualus. Įvertinti darbą sunkiau, nes negaliu įvertinti kiekvieno indėlio, tačiau mokymosi efektas geresnis. Nereikėjo aiškinti, kaip generuojami sveikieji skaičiai. Darbas porose vyko sklandžiau nei grupėmis. Kai kuriems šiek tiek pritrūko laiko sprendžiant individualias užduotis, apsidžiaugė, kad leidau baigti namuose ir pateikti padarytus darbus iki rytojaus.

8 PRIEDAS

Naudotos vertinimo ir įsivertinimo formos

Darbo grupėse vertinimas

1. Komandos pavadinimas: _____ Lyderis: _____

2. Įvertinkite 5 balų skalėje darbą grupėje:

Mano idėjos ir pasiūlymai nesulaukė tinkamo dėmesio.	1	2	3	4	5	Mano idėjos ir pasiūlymai sulaukė tinkamo dėmesio.
Nemanau, kad komandos lyderis domėjosi mano idėjomis.	1	2	3	4	5	Manau, kad komandos lyderis buvo labai susidomėjęs mano idėjomis.
Šioje komandoje nebuvo tinkamo bendradarbiavimo ir per mažai pagrįstų nutarimų.	1	2	3	4	5	Komanda puikiai bendradarbiavo ir priėmė pagrįstus sprendimus.
Komandos nariai neįsitraukė į juos dominančių klausimų aptarimą.	1	2	3	4	5	Komandos nariai įsitraukė į juos dominančių klausimų aptarimą.
Jaučiausi nepatogiai, kai komandoje teko kalbėti apie savo klaidas.	1	2	3	4	5	Komandoje jaučiausi patogiai, čia galėjau kalbėti apie savo padarytas klaidas.
Diskusijos mūsų komandoje nedavė jokių rezultatų.	1	2	3	4	5	Diskusijos mūsų komandoje davė rezultatus.
Mūsų komandos darbo ir kokybės lygis žemas.	1	2	3	4	5	Mūsų komandos darbo ir kokybės lygis aukštas.
Užduoties išspręsti nepavyko.	1	2	3	4	5	Užduotį išsprendėme puikiai.

3. Ką naujo sužinojau atlikdama(-as) užduotį?

4. Kas buvo sunkiausia atliekant užduotį ir kodėl?

5. Kokį pažymį parašysi sau? Kodėl?

6. Ar patiko dirbti komandoje?

7. Ką reikėtų patobulinti?

Pristatymų vertinimas

Įvertink kolegų pristatymus

Komanda										
	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Kiek naujos informacijos išgirdai?										
	Nesutinku		Iš dalies nesutinku		Neturiu nuomonės		Iš dalies sutinku		Visiškai sutinku	
Pristatymas buvo aiškus ir suprantamas										
Mano nuomone, tema atskleista										
Kas liko neaptarta?										

Ar patiko pristatinėti darbą? Kodėl?

Individualaus darbo įsivertinimas

Prašau įvertink, kurie iš darbo etapų Tau buvo sudėtingiausi. Sunumeruok darbus nuo paties lengviausio iki sunkiausio:

	Vieta
Užduoties analizė	
Algoritmo aptarimas	
Algoritmo užrašymas (schema, ar veiksmų seka)	
Algoritmo užrašymas programavimo kalba	
Testavimas ir derinimas	
Komentavimas ir paaiškinimai	
Vertinimas	

Kokį pažymį sau parašysi? Kodėl?

Ar patiko dirbti individualiai? Kodėl?

Kuri užduotis buvo sunkiausia? Kodėl?

9 PRIEDAS

Programavimui mokytis taikomų mokymosi metodų sąrašas

Kodas, mokymosi metodas arba veikla	Aprašymas, pastabos	Taikymas	Apribojimai	Besimokančiųjų atliekami veiksmai
MM01: Algoritmų ir programų vizualizavimas	Vizualizavimo priemonių taksonomijoje (Mayer, 1990), išskiriamos dvi kategorijos: 1) algoritmų vizualizavimo priemonės, 2) programų vizualizavimo priemonės. Vienas iš seniausių algoritmų vizualizavimo būdų – blokinių schemų braižymas. „Programų vizualizavimas suteikia galimybę programos tekstą arba programos veikimą pavaizduoti naudojant grafines metaforas“ (Mayer, Anderson, 1991; Moreno ir kt., 2004; Sorva, 2012). Programų vizualizavimo priemonės savo ruožtu skirstomos į programos teksto vizualizavimą ir programos veikimo dinamikos vizualizavimą. Programų veikimo dinamikos vizualizavimo priemonės sudaro sąlygas sekti programos kintamųjų reikšmes, iš eilės atliekant programoje numatytus veiksmus.	Blokinių schemų braižymas tinkamas kuriant užduoties atlikimo algoritmą, numatant testavimo reikšmes. Programų veikimo dinamikos vizualizavimo priemonės padeda geriau suprasti, kaip veikia programiniai konstruktai, kaip kinta kintamųjų reikšmės programos vykdymo metu. Animacija tinka aiškinant ciklus, rikiavimo algoritmus ir pan.	Metodas netinkamas tiems, kurie sunkiai priima vaizdinę informaciją ir / ar nesupranta grafinių metaforų.	Analizuoja algoritmų blokines schemas, aktyviai mąsto, kuria algoritmo grafinį vaizdą, stebi animacijas arba programų vizualizacijas.
MM02: Aiškumas	V. Rajecko (1997) teigimu, tai žodinis dėstymo metodas, kurio tikslas – formuoti sąvokas, dėsnius, taisykles, instruktuoti studentus. Tai nuoseklus, logiškas naujos mokomosios medžiagos dėstymas.	Gali būti taikomas skaitant paskaitas ar demonstruojant.	Dažnai studentai lieka pasyvūs stebėtojai.	Stebi ir klauso; klausinėja.
MM03: Darbas pagal instrukcijas „žingsnis po žingsnio“	Iš anksto parengiama užduotis ir nuoseklus planas, kaip ją atlikti, kartais vadinamas laboratoriniu darbu.	Taikomas įvedant naujas sąvokas ir programavimo kalbos konstrukcijas.	Netinkamas tiems, kurių programavimo gebėjimai geri.	Dirba pagal iš anksto aprašytas instrukcijas.

Kodas, mokymosi metodas arba veikla	Aprašymas, pastabos	Taikymas	Apribojimai	Besimokančiųjų atliekami veiksmai
MM04: Mokymasis bendradarbiaujant	Darbo grupėse metodas, kai grupės nariai turi vieną bendrą tikslą ir yra skatinami kurti vieną bendrą rezultatą. Visi grupės nariai gali būti vertinami bendrai arba atskirai, į vertinimo procesą įtraukiant visus grupės narius.	Tinkama, kai siekiama atlikti sudėtingesnes užduotis, kurios gali turėti keletą sprendimo variantų.	Sudėtinga tinkamai suformuoti darbo grupes, esant dideliame studentų skaičiui grupėje. Darbui grupėse reikalinga atitinkama mokymosi aplinka. Ne visi studentai prisiima atsakomybę už savo ir grupės narių mokymąsi.	Jei prisiima atsakomybę, aktyviai mokosi, dalijasi patirtimi, bendrauja, diskutuoja, bendradarbiauja, vertina vieni kitų ir įsivertina savo darbą.
MM05: Problemų sprendimas	Studentai, dėstytojui padedant, mokosi spręsti problemas. Problemų mokymosi metodas koncentruojasi ties sudėtinga problema, neturinčia vieno teisingo atsakymo. Dažniausiai studentai dirba bendradarbiaudami grupėse, siekdami nustatyti, ką jie turi sužinoti, kad išspręstų problemą.	Taikoma skatinant studentus įsitraukti į savivaldų mokymąsi, pritaikyti savo naujas žinias problemai spręsti ir reflektuoti savo mokymąsi bei naudotų strategijų efektyvumą.	Sprendžiama problema turi atitikti besimokančiųjų programavimo gebėjimų lygį.	Aktyviai mokosi grupėse arba individualiai, analizuoja problemą, ieško sprendimo būdų, priima sprendimus, sprendžia problemas.
MM06: Demonstravimas	Demonstravimo esmė: parodyti, kaip atlikti užduotį, kad vėliau studentas galėtų užduotį atkartoti. Jeigu demonstruojama sprendžiama programavimo užduotis, reikėtų rodomus veiksmus analizuoti ir skaidyti į atskirus fragmentus, juos komentuoti, užduoti probleminius klausimus. Kartais verta užduoties sprendimo demonstravimą patikėti geresnius programavimo įgūdžius turinčiam besimokančiajam.	Taikomas įvedant naujas programavimo sąvokas ar sintaksės konstrukcijas.	Dažnai studentai lieka pasyvūs stebėtojai.	Stebi, bando atkartoti stebimus veiksmus, klausinėja.

Kodas, mokymosi metodas arba veikla	Aprašymas, pastabos	Taikymas	Apribojimai	Besimokančiųjų atliekami veiksmai
MM07: Diskusija	Organizuojant diskusiją būtina iš anksto numatyti temą ir siekiamus tikslus. Diskusija turi būti kontroliuojama ir plėtojama, užduodant klausimus, skatinant studentus aktyviai išsakyti. Diskusijoms būtina sukurti baimės nekeliančią aplinką, skatinti visus dalyvauti. Jos gali būti organizuojamos ir virtualiai, pasinaudojant virtualiąją mokymosi aplinką, socialinio tinklo grupe ir pan.	Taikoma, kai mokymosi tikslas – aptarti paskaitoje pateiktus teiginius, sprendžiamą problemą (atlikti užduoties analizę) ar išsiaiškinti ir panaudoti studentų turimą patirtį.	Diskusijos eiga ir rezultatas labai priklauso nuo dalyvių gebėjimo reikšti savo mintis, argumentuoti, gebėjimo išklausti kitus. Suteikia galimybių reikšti aktyviems, o pasyvūs lieka nuošalyje. Diskusija gali tapti nenaudinga.	Aktyviai diskutuoja, klausosi kitų.
MM08: Programavimas porose	Šio metodo esmė: užduotis atliekama dviese prie vieno kompiuterio. Abu dalyviai diskutuodami kuria ir testuoja tą pačią programą. Vienas iš jų yra vedėjas – jis sėdi prie klaviatūros ir rašo, kitas yra tyrinėtojas – jis stebi vedėjo darbą, stengiasi pastebėti potencialias klaidas, patarinėja, siūlo alternatyvas. Taikant šį metodą, užduotis atliekama greičiau, padaroma mažiau klaidų, formuojamas teigiamas požiūris į bendradarbiavimą.	Taikomas siekiant pagerinti silpnesniųjų programavimo kalbos sintaksės žinias, algoritmavimo, programų testavimo, derinimo ir bendravimo bei bendradarbiavimo gebėjimus.	Parenkant poras būtina atsižvelgti į studentų programavimo išgūdžius. Labai skirtingo lygio poros bendradarbiavimas gali neduoti laukiamos naudos (Zacharis, 2011).	Jei prisiima atsakomybę, aktyviai mokosi, ieško informacijos, ją analizuoja, dalijasi patirtimi, bendrauja, diskutuoja, bendradarbiauja, vertina vieni kitų ir įsivertina savo darbą.
MM09: Durstinys	Durstinio (anglų k. <i>Jigsaw</i>) metodą pirmasis pasiūlė E. Aronsonas (JAV). Tai metodas, kuris remiasi bendradarbiaujančių grupių principu ir konstruktyvizmo teorija. Metodas buvo sugalvotas kaip kontaktinio (anglų k. <i>face-to-face</i>) mokymosi metodas, tačiau daugelis mokslininkų aprašo, kad jį galima sėkmingai pritaikyti nuotoliniam mokymui (Blocher, 2005; Vargas ir kt., 2011). Taikant durstinio metodą besimokantieji padeda integruoti žinias, sukauptas iš įvairių šaltinių (Aronson, 2000).	Taikant šį metodą lavinami skaitymo, kalbėjimo, bendradarbiavimo, reflektatyviojo mąstymo, problemų sprendimo išgūdžiai. Tinka didesnės apimties programavimo užduotims, kurios gali būti padalijamos į smulkesnes dalis.	Verta taikyti tuomet, kai besimokantieji turi aiškinių išgūdžių, o jų programavimo gebėjimai yra panašaus lygio.	Jei prisiima atsakomybę, aktyviai mokosi, dalijasi patirtimi, bendrauja, diskutuoja, bendradarbiauja, ieško informacijos, ją analizuoja, sprendžia problemas, vertina vieni kitų ir įsivertina savo darbą.

Kodas, mokymosi metodas arba veikla	Aprašymas, pastabos	Taikymas	Apribojimai	Besimokančiųjų atliekami veiksmai
	<p>Metodo esmė: besimokantieji suskirstomi į grupes, jiems skiriama užduotis, kurią jie analizuoja ir padalija į mažesnes dalis. Po to pirminės grupės išsiskirsto į taip vadinamąsias ekspertų grupes, kurios turi mažesnes, bet skirtingas vienos pagrindinės užduoties dalis. Išsprendus atskiras užduoties dalis ekspertų grupės išardomos ir grįžta į savo pirmines grupes. Į kiekvieną grupę sugrįžta po ekspertą iš kiekvienos ekspertų grupės. Grupės turi išspręsti pagrindinę užduotį, pasinaudodamos ankstesnėse grupėse įgyta patirtimi.</p>			
<p>MM10: Programos teksto skaitymas (analizė) ir veikimo sekinimas</p>	<p>Taikant šį mokymosi metodą studentui pateikiamas jau sukurtos programos tekstas be komentarų ir užduoties, kurią programa sprendžia, sąlygos. Skaitydamas programos tekstą ir jį analizuo-damas studentas turi išsiaiškinti, kokią užduotį sprendžia skaitoma programa. Paprastesnės šio metodo realizacijos – pakomentuoti kiekvieną programos teksto eilutę arba sekti, kaip programos vykdymo metu kinta kintamųjų reikšmės. Siekiant tobulinti programavimo kalbos sintaksės žinias, gali būti pateikiamas klaidingas programos tekstas su užduotimi ištaisyti klaidas.</p> <p>Programos teksto skaitymo metodas gali būti taikomas naudojantis dėlionės principu. Programos tekstas padalijamas į dalis (žodžiai nedalijami), kurios yra sumaišomos. Studentai turi surinkti programos tekstą iš detalių.</p>	<p>Naudojama programavimo kalbos sintaksei mokytis arba algoritmo veikimo principams išsiaiškinti. Tačiau atsižvelgiant į tai, kad šiuolaikinis programuotojas dažnai dirba komandoje, kartais tęsia kitų pradėtą darbą, jam svarbu gebėti skaityti ir suprasti kitų parašytas programas. Todėl mokantis programavimo reikia nepamiršti ugdytis gebėjimą skaityti ir suprasti kitų parašytas programas.</p>	<p>Yra žinoma, kad „gebėjimas parašyti programą ir gebėjimas ją skaityti bei sekti jos veikimą koreliuoja silpnai 0,5–0,6 ($p < 0,01$). Stipresnė koreliacija tarp programos sekimo ir rašymo pastebima studijuojant ciklinius algoritmus“ (Mannila, 2007; Lopez ir kt., 2008).</p>	<p>Aktyviai mokosi, skaito ir analizuoja pavyzdžius, ieško informacijos, ją analizuoja, keičia pavyzdžius, juos derina, testuoja, stebi pakeitimų rezultatus.</p>

Kodas, mokymosi metodas arba veikla	Aprašymas, pastabos	Taikymas	Apribojimai	Besimokančiųjų atliekami veiksmai
MM11: Individualus darbas	Individualiai atliekamos programavimo užduotys, būtinos ugdatant programavimo gebėjimus. Studentas turi išmokti įveikti visus programavimo užduoties etapus. Individualiai atliekamo darbo sėkmę lemia tinkamai pasirinktas savarankiškumo laipsnis. Dėstytojas gali gana primygtinai stebėti studentą ir jam vadovauti arba leisti jam dirbti savarankiškai, mažai jį prižiūrėti ir konsultuoti.	Taikomas tada, kai mokymosi tikslas – skatinti individualias kūrybines besimokančiojo galias mąstyti, ieškoti problemos sprendimo būdų.	Netinka, kai besimokančiojo programavimo gebėjimai yra silpni.	Aktyviai mokosi, mąsto, analizuoja, ieško informacijos, ją lygina ir analizuoja, kuria algoritmą, užrašo jį programavimo kalbos sakiniais, derina ir testuoja.
MM12: Minčių lietus	Minčių lietus – tai kūrybinis aktyvaus mokymosi metodas. Per trumpą laiką surenkama daug gerų, įdomių ir kartais originalių idėjų. Kaip teigia M. Teresevičienė (1999), tai yra „intensyvaus idėjų generavimo periodas, kurio metu sukuriamos idėjos ar galimi problemų sprendimo variantai yra svarbesni nei jų kokybė“ (Teresevičienė, 1999). Visos grupės narių išsakytos mintys užrašomos. Po tam tikro sutarto laiko visų išsakytos mintys analizuojamos ir priimamas galutinis sprendimas, padaromos išvados.	Taikomas atliekant programavimo užduoties analizę, pristatant naują temą. Taikant šį metodą skatinamas kūrybinis mąstymas.	Suteikia galimybių reikštis aktyviems, o pasyvūs lieka nuosalyje.	Aktyviai išsako savo mintis, klausosi kitų, apibendrina, daro išvadas, priima sprendimus.
MM13: Paskaita	Mikrolygmeniu suprantamas kaip pasyvus mokymosi metodas, kai lektorius, tam tikros srities ekspertas, perduoda informaciją žodžiu, o studentai klausosi, priima informaciją. Paskaitą sudaro įžanga, dėstymas ir išvados.	Taikomas tada, kai mokymosi tikslas – suteikti informacijos, nes kitais būdais mokymosi medžiaga nėra prieinama.	Netinka, kai dėstoma medžiaga sudėtinga, abstrakti, o mokymosi tikslas – suprasti sudėtingesnius pažinimo procesus: analizę, sintezę, vertinimą.	Klausosi ir įsitemina.

Kodas, mokymosi metodas arba veikla	Aprašymas, pastabos	Taikymas	Apribojimai	Besimokančiųjų atliekami veiksmai
Įtraukianti (interaktyvioji) paskaita	Siekiant didinti studentų aktyvumą paskaitos metu taikomos įvairios strategijos. Gali apimti daug ir įvairių metodų, kuriuos taikant siekiama įtraukti studentus į aptariamus klausimus ar problemą. Labai naudinga paskaitą pradėti sužadinimu, iškeliant probleminių klausimų. Taikant probleminio mokymo strategijas ugdomas kūrybinis ir kritinis mąstymas.	Taikoma tose aukštesiose mokyklose, kuriose populiarus kontaktinio darbo forma – paskaita, o dalyko tikslai – ugdyti gebėjimą diskutuoti, mąstyti, lyginti, daryti išvadas, vertinti.	Apribojimai susiję su interaktyviosios paskaitos metu taikomais metodais	Klausosi ir įsitemina, aktyviai klausinėja, diskutuoja, kelia ir sprendžia problemas.
MM14: Lektūra	Savarankiško mokymosi metodas. Skaitoma dėstytojo rekomenduota mokytojų literatūra.	Naudojama, kai mokymosi tikslas – pagilinti tam tikros srities žinias.		Skaitymas
MM15: Pristatymai	Individualiai ar grupėse atliktų darbų pristatymas. Studentai dalijasi patirtimi su bendramoksliais naudodamiesi savo parengta vaizdine medžiaga (vaizdo įrašu, skaidrių peržiūra, pateikdami įvykių suvestinę lentoje, organizuodami viešą diskusiją arba kt.). Diskutuoja ir vertina (įsivertina) pristatomus darbus (Brew, 2003).	Naudojama individualiai ir grupėse atliktiems darbams pristatyti ir įsivertinti.	Besimokančiųjų patiriami sunkumai: negeba minčių išreikšti žodžiu, nevaldo jaudulio, patiria viešo kalbėjimo baimę.	Viešai pristato savo darbo rezultatus.
MM16: Savikontrolės testas	Nors tai labiau įsivertinimo metodas, tačiau kai jis gerai organizuotas, suteikiantis išsamų grįžtamąjį ryšį, gali būti taikomas ir kaip mokymosi metodas. Įgūdžiai, įgyjami testavimo metu, padeda mokytiis kryptingai, susikaupti, skatinama savotiškai varžytis su savimi. Privalumas – gerai suprojektuotas grįžtamasis ryšys.	Tinkamas įsivertinanti ir įtvirtinanti programavimo sąvokų ir programavimo kalbos sintaksės žinias.	Laikant testus reikia gana detaliai žinoti mokymo medžiagą.	Sprendžia testo klausimus, aktyviai mąsto.

Kodas, mokymosi metodas arba veikla	Aprašymas, pastabos	Taikymas	Apribojimai	Besimokančiųjų atliekami veiksmai
MM17: Sąvokų žemėlapis	Sąvokų žemėlapis (arba kitaip koncepcijų žemėlapis) – tai konstruktyvaus mokymosi metodas-schema, kurią naudojant vaizdžiai pateikiamas supratimas apie objektus, įvykius, reiškinius arba tam tikrą jų dalį. Metodas remiasi mokymosi teorija, kurioje teigiama, kad anksčiau įgytos žinios lemia tolimesnį sėkmingą mokymąsi (Ausubel, 2000).	Tinkamas, kai mokymosi tikslas – suprasti sąvokų ir procesų ryšius. Ugdomas gebėjimas pereiti nuo įvairių sąvokų bei procesų prie jų grafinių vaizdų. Padeda integruoti naują informaciją į jau egzistuojančią žinių struktūrą, stebėti savo mokymąsi ir nustatyti žinių spragas. Padeda susisteminti ir pertvarkyti mintis. Šį metodą galima taikyti vienam ir grupėmis.	Metodas netinkamas tiems, kurie sunkiai priima vaizdinę informaciją ir / ar nesupranta grafinių metaforų.	Mąsto, analizuoja, ieško ryšių tarp sąvokų, piešia sąvokų grafinių vaizdų, identifikuoja trūkstamas žinias.
MM18: Vaizdo pamokos	Tikslingai mokytojo parengtos vaizdo pamokos gali būti naudojamos taikant atvirkštinės klasės metodą (anglų k. <i>Flipped Classroom</i>), kai studentai ruošiasi paskaitai peržiūrėdami naują teorinę medžiagą (pvz., parengtą vaizdo pamoką), o per pratybas aptaria iškilusius neaiškumus ir įtvirtina naują medžiagą (Milman, 2012; Davies, Dean, Ball, 2013). Analizuoti vaizdo pamokų naudojimo skirtumai, kai programuoti mokomasi kontaktiniu ir nuotoliniu būdais (Sharp, Schultz, 2013).	Dažniausiai naudojamos kaip savarankiškų studijų metodas arba studijuojantiems virtualiai siūlomas konceptualizavimo etape.	Studentai lieka pasyvūs stebėtojai. Dažnai vaizdo medžiaga neatitinka besimokančiojo poreikių.	Stebi ir įsimena.

Kodas, mokymosi metodas arba veikla	Aprašymas, pastabos	Taikymas	Apribojimai	Besimokančiųjų atliekami veiksmai
MM19: Vikis	Pagal apibrėžtį vikis (anglų k. <i>wiki</i>) yra interneto svetainių rūšis, kur lankytojai gali ne tik papildyti svetainės turinį kaip internetiniuose forumuose, bet ir jį taisyti. Sukurta daug intuityviai naudojamų vikio rašymo programų, o virtualiojoje mokymosi aplinkoje Moodle vikis įtrauktas į studentams siūlomų mokymosi veiklų sąrašą. Gali būti kuriamas bendras studentų grupės vikis. Atsižvelgiant į mokymosi metu susiformavusius poreikius, sukuriama struktūruota mokomoji medžiaga (vikis) gali būti naudingesnė nei dėstytojo parengtas konspektas ar mokomoji knyga.	Taikomas kaip rašto darbo metodas, mokantis programavimo sąvokų.	Gali būti sukaupta ne visa, nenuosekli mokomoji medžiaga.	Aktyviai mokosi, ieško informacijos, ją analizuoja, lygina, apibendrina, užrašo, sieja su kitais vikio puslapiais.
MM20: Žaidimų programavimas	Tai mokymosi metodas, kai paprastos, interaktyviosios grafinės programos integruojamos į įvadinio programavimo kursą. Abstrakčių programavimo sąvokų mokoma tyrinėjant ir programuojant mažus žaidimus (Sung, Hillyard ir kt., 2010; Theodorou, Kordaki, 2010). Akcentuojamas ne žaidimo sukūrimas, o supratimas, kokiais programavimo konceptais paremtas žaidimo veikimas. Studentai sugaišta gerokai daugiau laiko, kol supranta ir atlieka žaidimų užduotis. Sėkmingai atlikus užduotį stiprėja besimokančiųjų motyvacija.	Tokį metodą galima taikyti projektuose darbuose.	Pradedančiųjų programavimo gebėjimams ugdytis nelabai tinkamas, nes gali neapimti visų programavimo sąvokų.	Aktyviai mokosi, mąsto, ieško informacijos, ją lygina ir analizuoja, kuria algoritmą, užrašo jį programavimo kalbos sakiniais, derina ir testuoja.

Kodas, mokymosi metodas arba veikla	Aprašymas, pastabos	Taikymas	Apribojimai	Besimokančiųjų atliekami veiksmai
MM21: Varžybos	Tai probleminis mokymosi metodas. Studentams suformuluojama probleminė užduotis, o įvertinami tik pirmieji teisingai atlikę užduotį.	Taikoma individualizuojant mokymąsi, kai grupėje yra labai skirtingus programavimo gebėjimus turinčių studentų. Ši veikla dažniausiai skiriama tiems, kurių programavimo gebėjimai geri ir labai geri.	Užduotys turi atitikti besimokančiųjų programavimo gebėjimus. Svarbu, kad varžybose dalyvautų lygiaverčiai varžovai.	Aktyviai mokosi, mąsto, analizuoja, ieško informacijos, ją lygina ir analizuoja, kuria algoritmą, užrašo jį programavimo kalbos sakiniais, derina ir testuoja.

Veiklos tyrimo metu analizuotų metodų sąrašas

Kodas	Mokymosi metodai
MM01	Algoritmų ir programų vizualizavimas
MM04	Darbas grupėse bendradarbiaujant
MM05	Problemų sprendimas
MM06	Demonstravimas
MM07	Diskusija
MM08	Programavimas porose
MM09	Durstinys
MM10	Programos teksto skaitymas ir veikimo sekimas
MM11	Individualus darbas
MM12	Minčių lietus
MM13	Paskaita
MM15	Pristatymai
MM16	Savikontrolės testas
MM17	Sąvokų žemėlapis
MM18	Vaizdo pamoka
MM21	Varžybos

10 PRIEDAS

Mokymosi stilių ir mokymosi metodų dermė

Konceptualizacija				
Mokymosi metodai	A	M	T	P
MM01: Algoritmų ir programų vizualizavimas		x	x	
MM02: Aiškinimas		x	x	
MM06: Demonstravimas		x	x	x
MM07: Diskusija	x	x	x	x
MM09: Durstinys	x	x		
MM10: Programos teksto skaitymas ir veikimo sekimas		x		x
MM12: Minčių lietus	x	x		
MM13: Paskaita			x	
MM14: Lektūra		x	x	
MM15: Pristatymai	x			
MM16: Savikontrolės testai				x
MM17: Sąvokų žemėlapis		x	x	
MM18: Vaizdo pamoka		x	x	
MM19: Vikis			x	x
Konstravimas				
Mokymosi metodai	A	M	T	P
MM03: Darbas pagal instrukcijas „žingsnis po žingsnio“				x
MM04: Darbas grupėse bendradarbiaujant	x	x		
MM05: Problemų sprendimas	x		x	x
MM08: Programavimas porose	x	x		
MM09: Durstinys	x	x		
MM11: Individualus darbas			x	x
MM20: Žaidimų programavimas	x			x
MM21: Varžybos	x			
Dialogas				
Mokymosi metodai	A	M	T	P
MM07: Diskusija	x	x	x	x
MM16: Savikontrolės testas				x
MM11: Individualus darbas			x	x

11 PRIEDAS

Mokymosi stilių nustatymo klausimynas (LSQ)

Ši anketa skirta tam, kad sužinotume, kokiam(-iems) mokymosi stiliui(-iams) Jūs teikiate pirmenybę. Jeigu Jūs labiau sutinkate negu nesutinkate su teiginiu – šalia jo pažymėkite varnelę (✓). Jeigu Jūs labiau nesutinkate negu sutinkate su teiginiu – padėkite kryžiuką (X).

	Teiginys
	1. Aš turiu tvirtus įsitikinimus apie tai, kas yra teisinga ir neteisinga, gera ir bloga.
	2. Dažnai veikiu nesvarstydamas, kokios gali būti pasekmės.
	3. Aš esu linkęs spręsti problemas žingsnis po žingsnio (planingai).
	4. Aš manau, kad oficialios procedūros ir elgesio taisyklės riboja žmones.
	5. Aš turiu reputaciją žmogaus, kuris sako, ką galvoja, paprastai ir tiesiai.
	6. Dažnai patiriu, kad jausmais grindžiami veiksmai yra tokie pat teisingi (patikimi), kaip ir tie, kurie yra gerai apmąstyti ir išanalizuoti.
	7. Man patinka toks darbas, kurį atlikdamas turiu laiko išsamiai pasirengti ir (viską) įgyvendinti.
	8. Aš dažnai klausiu žmonių, kodėl jie vienaip ar kitaip galvoja.
	9. Man svarbiausia yra tai, ar kas nors pasiteisina praktikoje.
	10. Aš nuolat ieškau naujų patyrimų.
	11. Išgirdęs apie naujas idėjas ar metodus, aš iškart imu nagrinėti, kaip būtų galima juos praktiškai pritaikyti.
	12. Man labai svarbi savidrausmė, pvz.: laikytis dietos, reguliariai mankštintis, laikytis nustatytos dienotvarkės ir t. t.
	13. Aš didžiuojuosi, kad kruopščiai atlieku darbą.
	14. Aš geriausiai sutariu su logiškais, analitiškai mąstančiais žmonėmis ir ne taip gerai – su spontaniškais ir neracionaliais žmonėmis.
	15. Aš išsamiai išanalizuoju (visą) prieinamą informaciją ir vengiu skubotų išvadų.
	16. Aš mėgstu priimti sprendimą tik kruopščiai apsvarstęs daugelį galimybių.
	17. Mane labiau traukia naujoviškos ir neįprastos idėjos, o ne praktiškos.
	18. Man nepatinka blogai organizuoti dalykai, man labiau patinka veikti pagal logišką / aiškų modelį.
	19. Aš priimu ir laikausi nurodytų procedūrų ir taisyklių tol, kol jos man atrodo veiksmingos užduočiai atlikti.

	Teiginys
	20. Man patinka susieti savo veiksmus su bendraisiais principais.
	21. Diskusijose mėgstu iš karto pereiti prie dalyko esmės.
	22. Aš esu linkęs santūriai, oficialiai bendrauti su žmonėmis, su kuriais aš dirbu.
	23. Man prie širdies iššūkis ryžtingai imtis ko nors naujo ir kitoniško.
	24. Gėriusi smagiais, spontaniškais žmonėmis.
	25. Prieš darydamas išvadą aš smulkmeniškai išnagrinėju kiekvieną detalę.
	26. Man sunku sugalvoti idėjų (ką nors sumanyti) pagautam impulso.
	27. Aš (tvirtai) tikiu, kad reikia iš karto eiti prie reikalo.
	28. Aš stengiuosi nedaryti skubotų išvadų.
	29. Man labiau patinka turėti kaip galima daugiau informacijos šaltinių – kuo daugiau yra informacijos, kurią reikia apmąstyti, tuo geriau.
	30. Mane paprastai erzina lengvabūdiški žmonės, kurie nepakankamai rimtai žiūri į reikalus.
	31. Prieš išsakydamas savo požiūrį, aš pirma išklausau kitų žmonių nuomonę.
	32. Aš esu linkęs atvirai pasakyti, kaip jaučiuosi.
	33. Diskusijose mėgstu stebėti kitų dalyvių manevrus.
	34. Man labiau patinka reaguoti į įvykius spontaniškai ir lanksčiai, negu iš anksto viską suplanuoti.
	35. Mane traukia tokie specialūs metodai kaip tinklo analizė, proceso struktūrinės schemos, programų skaidymas, nenumatytų atvejų analizė ir t. t.
	36. Nerimauju, kai turiu skubiai atlikti darbą, kad spėčiau atiduoti jį laiku.
	37. Aš esu linkęs vertinti kitų žmonių idėjas pagal tai, ar jos yra praktiškos.
	38. Ramūs, mąslūs žmonės mane verčia jaustis nejaukiai.
	39. Mane dažnai suerzina žmonės, kurie nori skubinti įvykius.
	40. Daug svarbiau yra mėgautis dabartimi negu galvoti apie praeitį arba ateitį.
	41. Aš manau, kad sprendimai, priimti remiantis kruopščiai išanalizavus visą turimą informaciją, yra patikimesni negu tie, kurie buvo priimti remiantis tik intuicija.
	42. Galima sakyti, kad aš esu žmogus, siekiantis tobulumo.
	43. Diskusijose aš paprastai pateikiu daugybę spontaniškų idėjų.
	44. Susirinkimuose aš pateikiu praktiškų, realistiškų idėjų.
	45. Daugeliu atvejų taisyklės yra tam, kad jas laužytume.
	46. Man labiau patinka atsitraukti nuo situacijos ir apsvarstyti visas galimas perspektyvas.
	47. Aš dažnai pastebiu prieštaravimus ir trūkumus kitų žmonių argumentuose.
	48. Atsižvelgus į viską, aš kalbu daugiau negu klausausi.

	Teiginys
	49. Aš dažnai matau geresnius, labiau praktiškus būdus, kaip sutvarkyti reikalus.
	50. Aš manau, kad raštu pateiktos ataskaitos turi būti trumpos ir konkrečios (į temą).
	51. Aš tikiu, kad racionalus ir logiškas mąstymas turėtų imti viršų.
	52. Bendraudamas su žmonėmis, esu linkęs diskutuoti konkrečiais klausimais, o ne įsitraukti į socialines (bendro pobūdžio) diskusijas.
	53. Man patinka žmonės, kurie vertina dalykus realiai, o ne teoriškai.
	54. Diskusijose aš imu nekantrauti, kai nukrypstama nuo temos ar kalbama apie nesvarbius dalykus.
	55. Jeigu man tenka rašyti ataskaitą, aš visuomet prirašau daug juodraštinų variantų, prieš pateikdamas galutinę versiją.
	56. Man labai rūpi išbandyti dalykus, norint įsitikinti, ar jie tinkami praktikoje.
	57. Man patinka logiškai išmąstyti atsakymus.
	58. Aš mėgaujuosi, kai man tenka daug kalbėti.
	59. Diskusijose aš dažnai pastebiu, kad esu realistas, neleidžiu žmonėms nukrypti nuo temos ir sulaikau nuo neapgalvotų spėlionių.
	60. Prieš apsisprendamas mėgstu apsvarstyti daug galimų variantų.
	61. Diskusijose aš dažnai pastebiu, kad esu labiausiai nešališkas ir objektyvus.
	62. Diskusijose aš vengiu atkreipti dėmesį į save ir nesu linkęs lyderiauti ir daug kalbėti.
	63. Man patinka, kai aš galiu susieti šiuo metu atliekamus veiksmus su bendresniu įsivaizdavimu.
	64. Kai kažkas nepavyksta, aš mielai numoju į tai ranka ir nutariu, kad neverta dėl to labai išgyventi.
	65. Aš esu linkęs atmesti iki galo neapgalvotas, staiga į galvą šovusias idėjas, nes man jos atrodo nepraktiškos.
	66. Visuomet yra geriau pirma pamąstyti, o tik tada veikti.
	67. Atsižvelgus į viską, aš dažniau klausau negu pats šneku.
	68. Aš esu linkęs griežtai elgtis su žmonėmis, kuriems sunku logiškai mąstyti.
	69. Daugeliu atveju aš manau, kad tikslas pateisina priemones.
	70. Aš sutinku net užgauti žmonių jausmus, jei tik tai padės atlikti darbą.
	71. Mane varžo (gniuždo) reikalavimas turėti konkrečius tikslus ar planus.
	72. Paprastai priklausau prie tų, kurie į draugiją įneša gyvumo.
	73. Kad atlikčiau darbą, darau viską, ką, manau, esant tikslinga.
	74. Man greitai nusibosta planingas, kruopštus darbas.

	Teiginys
	75. Man labai patinka nagrinėti esmines prielaidas, principus ir teorijas, kuriais remiasi dalykai ir įvykiai.
	76. Man visuomet įdomu sužinoti, ką mano kiti žmonės.
	77. Man patinka, kai susirinkimai vyksta planingai, laikantis nustatytos darbotvarkės ir pan.
	78. Aš vengiu (šalinuosi) subjektyvių ar dviprasmiškų temų.
	79. Krizinėje situacijoje aš mėgaujuosi (dramine) įtampa ir jauduliu.
	80. Žmonės dažnai mano, kad aš nepaisau jų jausmų.

Violeta Jadzgevičienė

**MOKYMOSI STILIŲ IR MOKYMOSI METODŲ DERMĖ KAIP STUDENTŲ
PROGRAMAVIMO GEBĖJIMŲ UGDYMO VEIKSNYS**

Daktaro disertacija
Socialiniai mokslai, edukologija (07 S)

Redagavo Irutė Raišutienė
Maketavo Donaldas Petrauskas

SL 605. 23 sp. l. Tir. 10 egz. Užsak. Nr. 15-052
Išleido ir spausdino Lietuvos edukologijos universiteto leidykla
T. Ševčenkos g. 31, LT-03111 Vilnius
Tel. +370 5 233 3593, el. p. leidykla@leu.lt