

ŠIAULIŲ UNIVERSITETAS
INFORMATIKOS, MATEMATIKOS IR E. STUDIJŲ INSTITUTAS
INFORMATIKOS KATEDRA

Vidmantas Šarkys

Informatikos specialybės IV kurso nuolatinių studijų studentas

**Interaktyvios aplinkos mokymuisi programuoti Python kalba projektavimas
ir kūrimas**

**The Design and Development of the Interactive Environment for Learning
Programming in Python**

BAKALAURO DARBAS

Darbo vadovė: doc. dr. Sigita Turskienė

Recenzentas: lekt. dr. Valerijonas Dumskis

Šiauliai 2014/2015 m.m.

„Tvirtinu, jog darbe pateikta medžiaga nėra plagijuota ir paruošta naudojant literatūros sąraše pateiktus informacinius šaltinius bei savo tyrimų duomenis“

Vidmantas Šarkys

(parašas)

DARBO TIKSLAI IR UŽDAVINIAI

Tikslas:

Suprojektuoti ir sukurti interaktyvią mokymosi aplinką, skirtą mokytis programuoti *Python* kalba.

Uždaviniai:

- Atlikti populiarių virtualių mokymosi aplinkų apžvalgą.
- Suprojektuoti interaktyvią mokymosi aplinką, skirtą *Python* programavimo kalbos mokymuisi.
- Pagal parengtą projektą sukurti interaktyvią mokymosi aplinką.
- Atlikti interaktyvios mokymosi aplinkos testavimą ir derinimą.

Darbo vadovė _____
(vardas, pavardė, parašas)

TURINYS

DARBO TIKSLAI IR UŽDAVINIAI.....	3
TURINYS	4
ĮVADAS	5
1. Analitinė dalis.....	6
1.1. Virtualios mokymosi aplinkos samprata	6
1.2. Populiarių virtualių mokymosi aplinkų apžvalga.....	7
1.3. Kursų, skirtų mokytis <i>Python</i> programavimo kalbą, apžvalga	8
1.4. Aplinkos poreikių, reikalingų interaktyviam mokymuisi programuoti, analizė.....	9
1.5. <i>Python</i> programavimo kalbos pasirinkimo motyvacija	11
2. Projektinė dalis.....	12
2.1. Darbo vykdymo planas.....	12
2.2. Įrankių ir priemonių pasirinkimo analizė	12
2.3. Pradinis projekto aprašymas.....	14
3. Darbo eigos aprašymas	16
3.1. Darbų eigos grafai.....	16
3.2. Problemų ir jų sprendimų aprašymai ir pagrindimai	16
3.3. Galutinis projekto aprašymas	17
3.3.1. Programos kodo interpretavimas	18
3.3.2. Vartotojo kodo pateikties analizė vertinimas	18
3.3.3. Grafinė vartotojo sąsaja	19
3.3.4. Vartotojo duomenų bazė	20
3.3.5. Turinio prieinamumo valdymas	20
3.4. Darbo rezultatų analizė.....	22
3.5. Interaktyvios mokymosi aplinkos testavimas	22
3.6. Patarimai, pastebėjimai, rekomendacijos.....	22
4. Išvados	23
Literatūra ir informaciniai šaltiniai.....	24
Priedai	26
Interaktyvios aplinkos testavimas.....	26
Konferencijų programos ir sertifikatai	28
Anotacija	34
Summary	34

IVADAS

Mūsų kasdienybė vis labiau tampa neatsiejama nuo informacinių technologijų – jos darosi prieinamesnės vartotojui tiek kainos, tiek patogumo naudotis požiūriu. Neišvengiamai kyla ir programuotojų, plėtojančių programinį produktą, poreikis, o jų kiekiui teigiamą įtaką daro ir laisvas mokomojo turinio prieinamumas internetu.

Internete esanti mokomoji medžiaga suteikia galimybę mokytis programuoti visiškai savarankiškai ir nemokamai. Pradedantysis programuotojas, priklausomai nuo norimos išmokti programavimo kalbos bei įrankių, gali rasti jam aktualios medžiagos keliose skirtingose informacinėse svetainėse. Deja, kartais sudėtinga surasti kokybiškos medžiagos, įvadinių pavyzdžių, pratimų, todėl savarankiškas mokymasis tampa ganėtinai sudėtingas. Išeitis tokiu atveju – į programavimą orientuotos studijos aukštosiose mokyklose, tačiau jų kaina bei studijų programoje esančių papildomų modulių gausa gali atbaidyti žmogų.

Visas minėtas problemas gali išspręsti nuotolinio mokymosi aplinka, vienpusiškai orientuota į mokymąsį programuoti tam tikra kalba. Vartotojas, prisijungęs prie tokios aplinkos nuotoliniu būdu, vienoje vietoje gali rasti tiek reikiamą teorinę medžiagą, tiek praktines užduotis, gali patogiai įsitraukti į mokymosi procesą, kurio tempą taip pat gali reguliuoti pats.

Šio darbo tikslas – suprojektuoti ir sukurti interaktyvią mokymosi aplinką patogiam ir efektyviam bei savarankiškam *Python* programavimo kalbos mokymuisi.

Uždaviniai:

- Atlikti populiarių virtualių mokymosi aplinkų apžvalgą.
- Suprojektuoti interaktyvią mokymosi aplinką, skirtą *Python* programavimo kalbos mokymuisi.
- Pagal parengtą projektą sukurti interaktyvią mokymosi aplinką.
- Atlikti interaktyvios mokymosi aplinkos testavimą ir derinimą.

Bakalauro darbo rezultatai pristatyti mokslinėse-praktinėse konferencijose:

1. Dr. R. Pocevičienė, doc. dr. S. Turskienė, V. Šarkys - ***”IKT ir savarankiškas mokymasis”***. Šiuolaikinių informacinių ir ryšių technologijų panaudojimo įtaka ugdymo kokybei, Kretingos rajono pedagogų švietimo centras, 2015.
2. V.Šarkys - ***“Interaktyvi aplinka, skirta Python programavimo kalbos mokymuisi”***. Mokslas ir studijos 2015: teorija ir praktika, Šiaurės Lietuvos kolegija, 2015

1. ANALITINĖ DALIS

1.1. Virtualios mokymosi aplinkos samprata

Virtuali mokymosi aplinka (VMA) – tai sistema, skirta mokomosios medžiagos pateikimui elektroniniu būdu, mokymosi veikloms organizuoti, pedagogams bendrauti ir bendradarbiauti su besimokančiais. Tai aplinka, kurioje galima pasiekti mokymosi medžiagą, atlikti testus ir įvairias užduotis bei mokytis patogiu metu, tempu. Virtuali mokymosi aplinka – tai puiki priemonė ilgesnei mokomajai veiklai, ypač kai studentai turi galimybių mokytis tik nuotoliniu būdu [1].

Virtualiosios mokymosi aplinkos puikiai tinkamos naudoti įvairiose mokslo bei mokymo institucijose, kadangi leidžia ne tik įdomiai, patraukliai pateikti mokymosi medžiagą, tačiau ir padidinti šios medžiagos pasiekiamumą. Priklausomai nuo medžiagą pateikiančių pedagogų tikslų bei pateikiamos medžiagos specifikos, virtuali mokymosi aplinka gali būti naudojama tiek kaip pagalbinė tam tikro kurso ar paskaitos priemonė, tiek kaip ir visavertė nuotolinio mokymosi kurso realizacija.

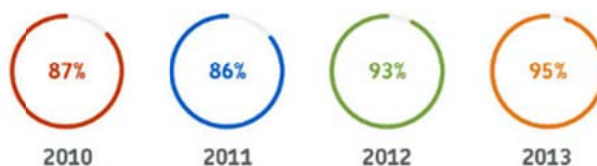
E. mokymės arba nuotolinės studijos taikomi įvairūs e. mokymosi teikimo būdai, kuriais pasiekiamas mokymosi proceso efektyvumas:

- Visiškai savarankiškas mokymosi procesas, kurio metu besimokantis asmuo gali mokytis be mokytojo, o tai leidžia personalizuoti mokymąsi pagal besimokančiojo poreikius;
- Mokomoji veikla padedant mokytojui;
- E. mokymasis bendraujant – mokymasis padedant mokytojui, kai į veiklą integruojamos virtualios klasės, konferencijos, pokalbių grupės, forumai ir kitos virtualios komunikacijos priemonės;
- Mobilus e. mokymosi procesas, kurio metu taikomos mobilios technologijos;
- Mišrus e. mokymasis – kartu naudojamos kelios iš aukščiau minėtų formų [2, 3].

1.2. Populiarių virtualių mokymosi aplinkų apžvalga

Pastaraisiais metais virtualaus mokymosi paklausa auga. Iš 500 didžiausių JAV įmonių (dar bendrai vadinamų *Fortuna 500*) net 41.7 proc. jų naudoja tam tikrą e. mokymosi formą savo darbuotojų kvalifikacijai kelti. Pažvelgę į vieno didžiausių Didžiosios Britanijos Surrey universiteto pateikiamą statistiką apie studentų pasitenkinimą virtualiu mokymusi, pastebime, kad e. mokymosi paklausa paskutiniu metu auga [4, 5]:

Surrey universiteto studentų pasitenkinimo virtualiu mokymusi rodikliai



1 pav. Surrey universiteto studentų pasitenkinimo e. mokymusi kilimas [5]

Kylant e. mokymosi technologijų populiarumui bei paklausai, natūralu, kad virtualių mokymosi aplinkų paklausa taipogi kyla. Iš šiuo metu populiarių virtualių mokymosi aplinkų galime išskirti šias: *Moodle*, *SAKAI*, *WebCT*, *FirstClass*, *Brightspace*. [3, 6].

Moodle yra VMA, skirta mokomojo turinio pateikimui, vartotojo progreso įvertinimui (naudojant testus) bei komunikacijai tarp studentų ir dėstytojų (pokalbių kambariai, forumai). *Moodle* nėra apribota naudoti vien programavimo mokymui – ji pritaikyta bet kokios srities mokymams, o tai šiai aplinkai suteikia universalumo bei skatina populiarumą visame pasaulyje. A. Rudaitis savo darbe yra aprašęs virtualios mokomosios aplinkos, skirtos mokytis programuoti *Python* kalba bei paremtos *Moodle* aplinka, kūrimą. Jis taip pat realizavo *Python* programavimo kalbai mokytis skirtą kursą *Moodle* aplinkoje [7, 8].

SAKAI yra akademinių institucijų bendruomenės kuriama virtuali mokymosi aplinka, skirta įvairaus pobūdžio mokymosi procesams palengvinti. Tai atviro kodo aplinka, prieinama virš 20 kalbų. Šios aplinkos funkcionalumas padeda planuoti, optimizuoti mokymosi procesą, bendrauti su kitais kurso dalyviais bei dėstytojais [9]. R. Mačiulskis išsamiai VMA *SAKAI* galimybes aprašė savo darbuose [10, 11].

WebCT (šiuo metu žinoma kaip *BlackBoard*) yra virtualaus mokymosi aplinka, kuri leidžia dėstytojams kurso metu naudotis tokiais įrankiais kaip diskusijų lentos, pašto sistemos, gyvi pokalbiai ir pan. *WebCT* laikoma pirmąją sėkmingą nuotolinių kursų realizacijos sistema pasaulyje. [12]

FirstClass yra kliento bei serverio pusėje veikianti grupinio darbo, el. pašto, virtualių konferencijų ir kt. įrankių sistema, veikianti Windows, Macintosh bei Linux operacinėse sistemose. *FirstClass* orientuojasi į aukštojo mokslo rinką, teikdama paslaugas dešimtyje didžiausių mokslo regionų JAV [13].

Brightspace (kompanijos *Desire2Learn* produktas) – tai nuotoliniam mokymuisi realizuoti skirtas įrankis, kuriame instruktoriaus kuriamas užduotis atlikinėja besimokantieji. Šiuo įrankiu realizuojamas mokymosi procesas tampa žaismingas, dinamiškas [14].

Lentelė 1. Populiarių mokymosi aplinkų palyginimas [10, 11, 12, 13, 14]

	Atviro kodo aplinka	Skirta tik programavimo mokymuisi	Galima naudotis naršyklėje	Mokytojas/dėstytojas yra būtinas dalyvis aplinkoje	Yra testai
<i>Moodle</i>	Taip	Ne	Taip	Taip	Taip
<i>SAKAI</i>	Taip	Ne	Taip	Taip	Taip
<i>WebCT</i>	Ne	Ne	Taip	Taip	Ne
<i>FirstClass</i>	Ne	Ne	Ne	Taip	Taip
<i>Brightspace</i>	Ne	Ne	Ne	Taip	Taip

Kaip pastebime iš 1 lentelės duomenų, nė viena iš populiariausių virtualių mokymosi aplinkų nėra išskirtinai orientuota į programavimo kursų kūrimą, todėl aplinką galima būtų naudoti arba tik mokymosi proceso organizacijos tikslais, arba reikėtų į aplinką integruoti papildomų įrankių, skirtų būtent programos kodo tikrinimui arba kitokiam vartotojo programavimo žinių tikrinimui. Taip pat galime pastebėti, jog visose aplinkose mokytojas/dėstytojas yra būtinas proceso dalyvis, t.y. nėra orientuojamasi į proceso automatizavimą. Taipogi matome, jog beveik visose aplinkose yra testų, skirtų vartotojo mokymosi progreso įvertinimui – galime daryti išvadą, kad tai yra labai svarbi mokymosi aplinkos dalis.

1.3. Kursų, skirtų mokytis *Python* programavimo kalbą, apžvalga

CodeAcademy. Leidžiama nemokamai mokytis *JavaScript*, *C#*, *Python*, *HTML* ir kt. kalbų interaktyviuose kursuose. Kiekviename kurse vartotojas rašo kodą pagal kairėje ekrano pusėje pateiktą užduotį, o dešinėje ekrano pusėje gauna pastabas apie kode esančias klaidas. Kursų langų apatiniame dalyje yra nuorodos į glaustą teorinės medžiagos puslapį bei dažniausiai užduodamų klausimų skiltį. Kursai realizuoti tik anglų kalba [15].

Masinis atviras internetinis kursas „Žaidimų programavimo įvadas Python kalba“.

Šis nemokamas atviras kursas, skirtas 7-11 kl. moksleiviams, leidžia ne tik susipažinti su *Python* programavimo kalbos subtilybėmis, bet ir praktiškai susidurti su žaidimų kūrimu. Kursas realizuotas naudojant *Moodle VMA*, realizuoti testai, kodo pateiktis tikrinama *Trinket* įrankiu, visa medžiaga bei užduotys pateikiama lietuvių kalba. Kurso peržiūros metu vartotojams pasiekama buvo tik labai nedidelė kurso medžiagos dalis, tačiau ateityje žadama, kad bus pateikiamas pilnas užduočių bei medžiagos komplektas [16].

Trinket – „*Hour of Python*“. Jau minėtas įrankis *Trinket* gali būti nesudėtingai pridedamas į bet kurį internetinį puslapį kaip įterpiamas html kodo fragmentas, tuomet leidžiama jame patogiai ir greitai vykdyti *Python* arba HTML kodą. Tiesa, naudojantis *Trinket* įrankiu yra realizuotas ir kursas „*Hour of Python*“, kuriame kelių interaktyvių pamokų metu siekiama vartotojui suteikti bazinių žinių apie *Python* programavimo kalbą. Visos užduotys pateikiamos tik anglų kalba [17].

1.4. Aplinkos poreikių, reikalingų interaktyviam mokymuisi programuoti, analizė

Nepriklausomai nuo to, ar žmogus programuoti mokosi savarankiškai, ar programavimo žinių semiasi tam tikroje švietimo įstaigoje, sėkmingam ir efektyviam mokymosi procesui būtina literatūra su programinio kodo pavyzdžiais. Tačiau net perpratus tam tikros programavimo kalbos sintaksę tenka susidurti su tam tikromis specifinėmis problemomis, kurios literatūriniuose šaltiniuose gali būti neapžvelgtos. Čia ypatingai pasitarnauja bei neretai didžiulio populiarumo susilaukia internetinės programavimo pamokų svetainės, taip pat – forumai bei pokalbių kambariai, kuriuose patyrę programuotojai gali padėti besimokantiejiems bei savo kolegoms atsakydami į tam tikrus praktinius klausimus [18, 19].

Atsižvelgiant į minėtus faktorius buvo nuspręsta projektuoti interaktyvią mokymosi aplinką, kurioje mokymosi proceso esmė būtų ne įsisavinti teorinę medžiagą bei bandyti įgytas žinias panaudoti praktiškai, tačiau atvirkščiai – daryti pateiktas užduotis laisvai eksperimentuojant, remiantis pateiktais pagalbinais pavyzdžiais bei teorine medžiaga, jei tam atsirastų poreikis. Tai leistų ne tik sutrumpinti mokymosi procesą, tačiau nuo pat pirmų jo akimirkų leistų besimokančiajam įsijausti į programavimo procesą, darant mokymąsi įdomesnį.

Tam, kad mokymosi aplinka atitiktų minėtą koncepciją, būtina realizuoti galimybę vertinti programos kodo įvestį bei pateikti vartotojui atsaką jam suprantama kalba bei pasufleruoti teisingą atsakymą, jeigu vartotojas nesugebėtų pateikti sąlygą tenkinančio programos kodo. Tai suteiks galimybę vartotojui bet kur ir bet kada (turint interneto ryšį bei internetinę naršyklę) pasiekti mokymosi aplinką ir be jokių papildomų įrankių visaverčiai mokytis programuoti pasirinkta programavimo kalba.

Be to, kaip jau minėta, virtualioje mokymosi aplinkoje turės būti pateikta ir teorinė medžiaga tam, kad, besimokančiajam prireikus, šis galėtų apie dominančius bei galbūt ne itin plačiai praktiškai apžvelgtus programavimo aspektus pasidomėti plačiau. Buvo nuspręsta pasinaudoti svetainės *www.coderland.lt* programavimo pamokų medžiaga medžiaga bei pateikti ją kaip teorinę medžiagą mūsų kuriamoje mokymosi aplinkoje.

Reikia, kad kiekvienas vartotojas turėtų galimybę sekti savo progresą, matyti atliktas užduotis, o tai leistų padaryti tik vartotojų paskyrų kūrimo galimybę. Kuriamoje aplinkoje nebus dėstytojų, mokytojų ar kt. labiau patyrusių asmenų, kurių dalyvavimo procese tikslas būtų peržiūrėti besimokančiojo progresą bei vertinti jo pateikiamus darbus. Atsisakant minėtų pareigų iš vienos pusės suteikiame galimybę dar labiau pagreitinti mokymosi procesą, iš kitos pusės įsipareigojame visiškai automatizuoti pateikiamų darbų analizę bei užtikrinti tikslų automatinį pateikiamų duomenų vertinimą.

Tad pagrindiniai reikalavimai, keliami kuriamai interaktyviai mokymuisi programuoti skirtai aplinkai yra šie:

- Mokymosi procesas grįstas praktine veikla, eksperimentavimu;
- Automatiškai tikrinama programos kodo pateiktis, automatinis pateikties vertinimas – realizuotas mokymosi proceso interaktyvumas;
- Naudojama *www.coderland.lt* svetainės teorinė medžiaga bei programos kodo pavyzdžiai;
- Visa medžiaga ir užduotys pateikiamos lietuvių kalba;
- Paskyrų sistema, leidžianti kiekvienam vartotojui sekti savo progresą.

1.5. *Python* programavimo kalbos pasirinkimo motyvacija

Python programavimo kalba pasižymi lengvai ir intuityviai suprantama sintakse, orientuota į programuotoją, kuris turi rašyti, skaityti bei suprasti programos kodą. *Python* tikslas yra šiek tiek priartinti programavimo kalbą prie žmonių kalbos, tam pasitelkiant elementarią skyrybos ženklų sistemą. *Python* savo sintaksėje turi vos 20 sakinių, todėl yra ypatingai greitai išmokstama bei lengvai suprantama nepriklausomai nuo operacinės sistemos, programavimo aplinkos ar kt. aplinkybių [20].

Be to, *Python* turi ir didžiulę standartinę biblioteką, kuri talpina daugybę naudingų funkcijų ir šablonų, kurie gali tiek palengvinti, tiek paspartinti programos kūrimo procesą. Atskirdami biblioteką nuo kalbos branduolio *Python* kūrėjai siekia išlaikyti pačią kalbą maksimaliai paprastą. Dėl šių priežasčių *Python* programavimo kalba jau keletą metų neužleidžia savo pozicijų populiariausių programavimo kalbų dešimtuose [21].

Virtualioje mokymuisi programuoti skirtoje aplinkoje turinys bus pateikiamas taip, kad net su programavimu anksčiau nesusidūręs asmuo, atlikinėdamas užduotis eilės tvarka, galėtų įgyti bent minimalių programavimo įgūdžių per kuo trumpesnę laiko tarpą. *Python* kalba dėl savo paprastos, intuityvios bei lengvai įsimenamos sintaksės, didžiulės standartinės bibliotekos bei paklausos šiandieninėje rinkoje atrodo kaip puikus pasirinkimas virtualioje aplinkoje, skirtoje besimokantiejiems programuoti.

2. PROJEKTINĖ DALIS

2.1. Darbo vykdymo planas

Lentelė 2. Darbų eigos grafas

	2015 vasaris				2015 kovas				2015 balandis			
1.												
2.												
3.												
4.												
5.												
6.												

1. Temos formulavimas, projekto įgyvendinimo galimybių analizė.
2. Aplinkos projektavimo ir kūrimo įrankių pasirinkimas.
3. Interaktyvios mokymosi aplinkos projektavimas.
4. Aplinkos kūrimas ir derinimas.
5. Aplinkos testavimas.
6. Darbo aprašymo rengimas.

2.2. Įrankių ir priemonių pasirinkimo analizė

Interaktyvios mokymosi aplinkos pranašumas lyginant su kitokio tipo aplinkomis – interaktyvus komunikavimas su vartotoju, t.y. atsakas į vartotojo veiksmus pateikiamas iš karto po užklauso. Tam, kad tai įgyvendintume, privalome automatizuoti vartotojo pateikčių analizavimo procesą arba rasti ir pasinaudoti įrankiu, leidžiančiu tai daryti. Pastarąjį variantą realizavo ir savo darbe aprašė J.Kairiūkštis. Jis atliko nuotolinio kompiliatoriaus *IdeOne* integraciją į programavimo pamokų svetainę, leisdamas vartotojui išbandyti svetainėje pateikiamus kodo pavyzdžius [22].

Neabejotinai pirmasis reikalavimas ieškomam įrankiui – *Python* programavimo kalbos palaikymas. *Python* yra interpretuojamoji kalba, todėl minėtam tikslui galime pasinaudoti nuotoliniu interpretatoriumi siųsdami jam užklauso ir analizuodami gaunamus rezultatus. Tiesa, yra ir keletas nuotolinių kompiliatorių, palaikančių užklauso *Python* kalba, tad renkantis įrankius galime peržvelgti ir juos. Ieškome nemokamo įrankio arba bent jau tokio, kuriuo būtų suteikiama nemokamo naudojimosi galimybė tam tikrą laiko tarpą.

Tam, kad savo mokomojoje aplinkoje galėtumėme naudotis išoriniu interpretatoriumi arba kompiliatoriumi, įrankis turi turėti viešą *API* – sąsają, per kurią galima būtų pasiekti nutolusios programos funkcionalumą. Šis reikalavimas kelia poreikį kitam – tiek *API*, tiek pats įrankio naudojimo procesas turi būti aprašyti, t.y. įrankis turi būti dokumentuotas tam, kad juo galima būtų sėkmingai naudotis su juo anksčiau nesusipažinusiam vartotojui. [23]

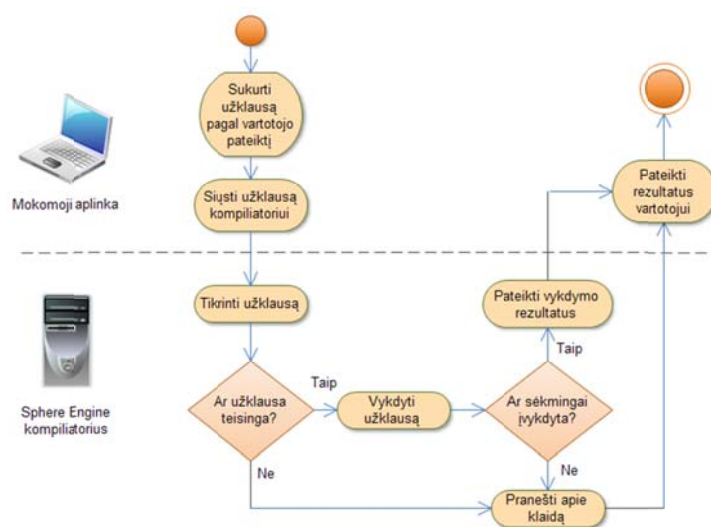
Pagal išvardintus poreikius buvo peržiūrėti 4 populiarius įrankiai, jų tinkamumas gali būti įvertintas tokia lentele:

Lentelė 3. Nutolusių Interpretatorių/kompiliatorių poreikių tenkinimo palyginimas [24, 25, 26, 27, 28].

	Galimybė naudotis nemokamai	<i>Python</i> kalbos palaikymas	Išorinis <i>API</i>	Dokumentacija	Išvaizdos pritaikymo galimybės
<i>Coding Ground</i>	+	+	-	+	-
<i>CodePad</i>	+	+	-	-	-
<i>Repl.it</i>	+	+	+	-	-
<i>Trinket</i>	+	+	-	-	-
<i>Sphere Engine</i>	+	+	+	+	+

Kaip pastebime, visus iškeltus poreikius tenkina nuotolinis kompiliatorius *Sphere Engine*. Šis įrankis nėra nemokamas, tačiau per mėnesį vienai vartotojo paskyrai suteikia 1000 nemokamų užklausų. [28]

Sphere Engine API pasiekiamas *SOAP* protokolu siunčiamomis žinutėmis. Užklausų siuntimo, vykdymo bei vykdymo rezultatų pateikimo procesą galima pavaizduoti tokia schema:



2 pav. Užklausų siuntimo, vykdymo bei vykdymo rezultatų pateikimo procesas

Kuriant interaktyvios mokymosi aplinkos grafinę vartotojo sąsają taip pat bus naudojamos web programavimo kalbos (*HTML*, *CSS*, *JavaScript*, *jQuery* biblioteka). Taip pat patogesnei bei išvaizdesnei vartotojo sąsajai sukurti bus naudojama *JavaScript* biblioteka *Ace*. Kadangi *Sphere Engine* naudoja vartotojo identifikavimą slaptažodžiu ir slaptažodžiu, užklausoje pateikiami duomenys juos siunčiant javascript kalba galėtų būti lengvai pasiekiami kiekvienam vartotojui. Dėl šios priežasties bendravimas su *Sphere Engine API* turėtų būti vykdomas serverio pusėje – bus naudojamos *PHP* funkcijos bei *AJAX* užklausos bendravimui tarp serverio bei vartotojo sąsajos.

Mokomosios aplinkos programavimui bus naudojami: *Sublime Text* kodo redaktorius, *Xampp v3.2.1* įrankių rinkinys serverio pusėje esančių funkcijų testavimui, bei *Edraw Max 6* bandomoji versija projekto diagramų rengimui.

2.3. Pradinis projekto aprašymas

Kuriamą interaktyvią mokymuisi programuoti skirtą aplinką sudarys keli esminiai elementai. Pirmiausiai vartotojui bus sudaryta galimybė įsisavinti *Python* programavimo kalbos niuansus atliekant pateiktas užduotis eilės tvarka. Užduotys bus suskirstytos teminiais skyreliais, kuriuose esančias užduotis ties bendra tematika, o užduočių eiliškumas priklausys nuo sudėtingumo – lengvesnės užduotys bus pateikiamos anksčiau, tokiu būdu leidžiant vartotojui racionaliai bei kryptingai įsitraukti į mokymosi procesą.

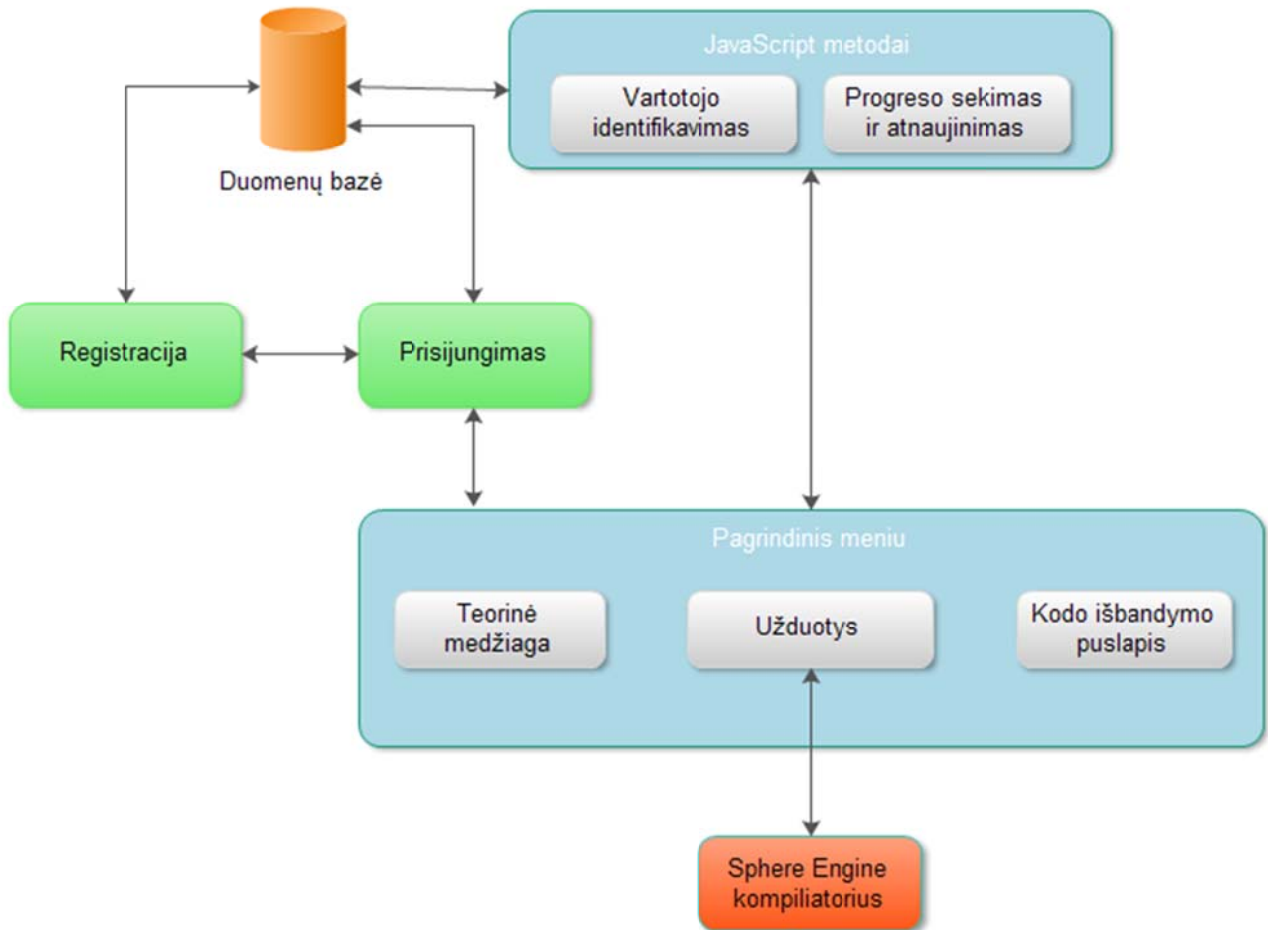
Vartotojui bus paliekama galimybė pačiam kontroliuoti kurso tempą – net atlikus pateiktą užduotį galima bus toliau likti toje pačioje pamokoje bei bandyti užduotį atlikti kitais metodais. Atsiradus sunkumų su užduoties atlikimu, vartotojas galės pasinaudoti viena arba dviem užuominomis, susijusiomis su praktiniu užduoties atlikimu.

Iškilus didesniems sunkumams atliekant užduotis bei atsiradus poreikiui labiau pasigilinti į programavimo kalbos subtilybes, vartotojui bus pateiktos teorinės medžiagos nuorodos, kurios leis pasiekti susijusią su pamokomis medžiagą www.coderland.lt svetainėje.

Vartotojas aplinkoje taip pat galės išbandyti savo programą – pateikti *Python* programos kodą bei gauti programos vykdymo *Sphere Engine* kompiliatoriumi rezultatus. Tai suteiks dar geresnes galimybes eksperimentavimui bei gilinimuisi į *Python* kalbos subtilybes. Be to, ši funkcija leis aplinka naudotis ir kaip patogiu *Python* kalbos interpretatoriumi.

Be abejo, vartotojas pirmiausiai turės būti identifikuotas tam, kad galėtų sekti būtent savo progresą bei pasiekti tik jam prieinamas pamokas (tam, kad galėtų pasiekti sekančią pamoką, turės įveikti ankstesnę), todėl reikalinga duomenų bazė su vartotojo paskyros bei progreso duomenimis, o taip pat – vartotojo identifikavimo ir progreso tikrinimo bei atnaujinimo galimybės.

Apibendrintai, interaktyvios mokymosi aplinkos projektą galime pavaizduoti taip:



3 pav. Interaktyvios mokymosi aplinkos projekto vizualizacijos UML interneto paslaugų diagrama

3. DARBO EIGOS APRAŠYMAS

3.1. Darbų eigos grafas

	2015 vasaris				2015 kovas				2015 balandis			
1.												
2.												
3.												
4.												
5.												
6.												

Lentelė 4. Darbų eigos grafas

7. Temos formulavimas, projekto įgyvendinimo galimybių analizė.
8. Aplinkos projektavimo ir kūrimo įrankių pasirinkimas.
9. Interaktyvios mokymosi aplinkos projektavimas.
10. Aplinkos kūrimas ir derinimas.
11. Aplinkos testavimas.
12. Darbo aprašo rengimas.

3.2. Problemų ir jų sprendimų aprašymai ir pagrindimai

Pirmoji problema, su kuria buvo susidurta – autentifikavimo keblumai dirbant su kompiliatoriumi *Sphere Engine*. Kaip jau buvo minėta anksčiau, šis įrankis suteikia dalinai nemokamą naudojimosi galimybę – 1000 nemokamų užklausų kompiliatoriui per mėnesį. Tačiau šis užklausų skaičius suteikiamas vartotojo paskyrai, todėl būtina vartotojo identifikacija kaskart siunčiant užklausą. Nors yra keli tipai klaidų, galinčių atsirasti užklauso siuntimo arba vykdymo metu, atlikus kompiliatoriaus pateikiamo atsako peržiūrą buvo pastebėtas klaidos kodas *AUTH ERROR*, kuris aiškiai ir vienareikšmiškai identifikuoja klaidą esant vartotojo paskyros autentifikavimo srityje.

Iš pradžių buvo ieškoma klaidos sprendimo kelių įrankio dokumentacijoje, tačiau toks atvejis ten plačiau nebuvo aprašytas. Internetė esantys pavyzdiniai kodo fragmentai, vaizduojantys veikiančią interakciją su *Sphere Engine* testavimo metu grąžindavo tą patį klaidos kodą, nors kodo autoriai tvirtai teikdavo, jog jų pateikiamos programos veikė bandymo metu.

Galiausiai buvo susisiepta su *Sphere Engine* įrankio administracija ir bandyta išsiaiškinti kilusios problemos priežastis bei sprendimo būdus, tačiau gautame atsakyme nebuvo konkrečiai atsakyta, kokia yra klaidos priežastis; buvo pasiūlyta išbandyti testavimui skirtą kodo fragmentą, esantį oficialioje įrankio kūrėjų svetainėje, kuris, anot administracijos, turėtų veikti visiems, teisingai įvedusiems identifikacinius duomenis.

Atsitiktinio testavimo metu slaptažodis buvo pakeistas į simbolių seką, sudarytą vien iš skaitmenų. Toks pakeitimas pasirodė esąs veiksmingas bei leido pradėti naudotis visais *Sphere Engine* kompiliatoriaus privalumais. Kūrėjams buvo nusiųstas klaidos aprašymo bei sprendimo būdo komentaras, kuris, tikėtina, ateityje padės išspręsti analogiškas problemas.

Kita problema atsirado bandant vartotojui suteikti galimybę rašyti taisyklingą kodo įvestį, kuri vėliau būtų siunčiama išoriniam kompiliatoriui. Neatsiejama *Python* programavimo kalbos dalis – indentacija, kuri leidžia formuoti funkcijas, ciklus ar formuoti kodo blokus iš kartu vykdomų kodo eilučių. Tradiciškai indentacija *Python* programavimo aplinkose formuojama klavišu TAB (kuriant indentaciją tarpais *Sphere Engine* praneša apie klaidą), tačiau naršyklėje šis klavišas turi kitokią standartinę paskirtį – nukreipti vartotoją į kitą įvesties elementą.

Buvo peržvelgti keli naršyklėms skirti plėtiniai, tokie kaip *Tabinta* (skirta naršyklei *Firefox*), kurie leidžia TAB klavišą naudoti teksto redaktoriams įprasta paskirtimi, tačiau bendro ir efektyvaus sprendimo rasti nepavyko. Galiausiai buvo pastebėta *JavaScript* biblioteka „*Ace*“, kurios paskirtis – sukurti html įvesties lauką, skirtą įvedamo kodo formatavimui. Šis įrankis leido tiksliai ir patogiai atlikti *Python* kodo įvestį, klavišo TAB paskirtis įvedimui skirtame lauke taipogi pasikeitė į tą, kuri būdingą teksto redaktoriuose.

3.3. Galutinis projekto aprašymas

Buvo sukurta interaktyvi mokymuisi programuoti *Python* kalba skirta aplinka, kurią sudaro trys esminiai elementai:

1. Programavimo užduotys su užuominomis bei realizuotas programinis kodo pateikčių tikrinimas;
2. Teorijos skiltis su nuorodomis į internetinių programavimo pamokų svetainę www.coderland.lt;
3. Skiltis, leidžianti vartotojui laisvai rašyti bei interpretuoti *Python* programos kodą, gauti programos vykdymo rezultatus.

3.3.1. Programos kodo interpretavimas

Praktinėse užduotyse *SOAP* protokolu siunčiamomis žinutėmis bendraujant su nuotoliniu kompiliatoriumi Sphere Engine, buvo realizuota interaktyvaus kodo interpretavimo galimybė. Serverio pusėje esančiame php faile *Sphere.php*, su kuriuo *AJAX* užklausomis komunikuoja mokomoji aplinka, aprašytas užklausų siuntimo *Sphere Engine* kompiliatoriui bei interpretuoto kodo rezultatų gavimo procesas. Jis iš esmės vykdomas 3 funkcijomis:

1. *CreateSubmission()* – sukuriamas siunčiamo kodo vykdymo užklausa. Kaip parametrai nurodomi vartotojo prisijungimo duomenys, programos kodas, programavimo kalba, bei vartotojo įvesties imitavimui skirti duomenys programos vykdymo metu;
2. *GetSubmissionStatus()* – peržiūrima kodo vykdymo būseną; tikrinama, ar užklausa vykdoma, ar jau įvykdyta;
3. *GetSubmissionDetails()* – gavus patvirtinimą, jog užklausa yra įvykdyta, naudojantis šia funkcija gaunami programos vykdymo rezultatai – programos išvesties tekstinė informacija arba įvykusios klaidos aprašymas.

3.3.2. Vartotojo kodo pateikties analizė ir vertinimas

Tiesa, gaunamas programos kodo vykdymo rezultatas nepateikia faktinės informacijos apie tai, ar vartotojo įvestis yra teisinga, ar atitinka užduoties reikalavimus. Siekiant analizuoti ir įvertinti kodo pateiktį kiekvienoje pamokoje naudojamos *JavaScript* kalba realizuotos funkcijos:

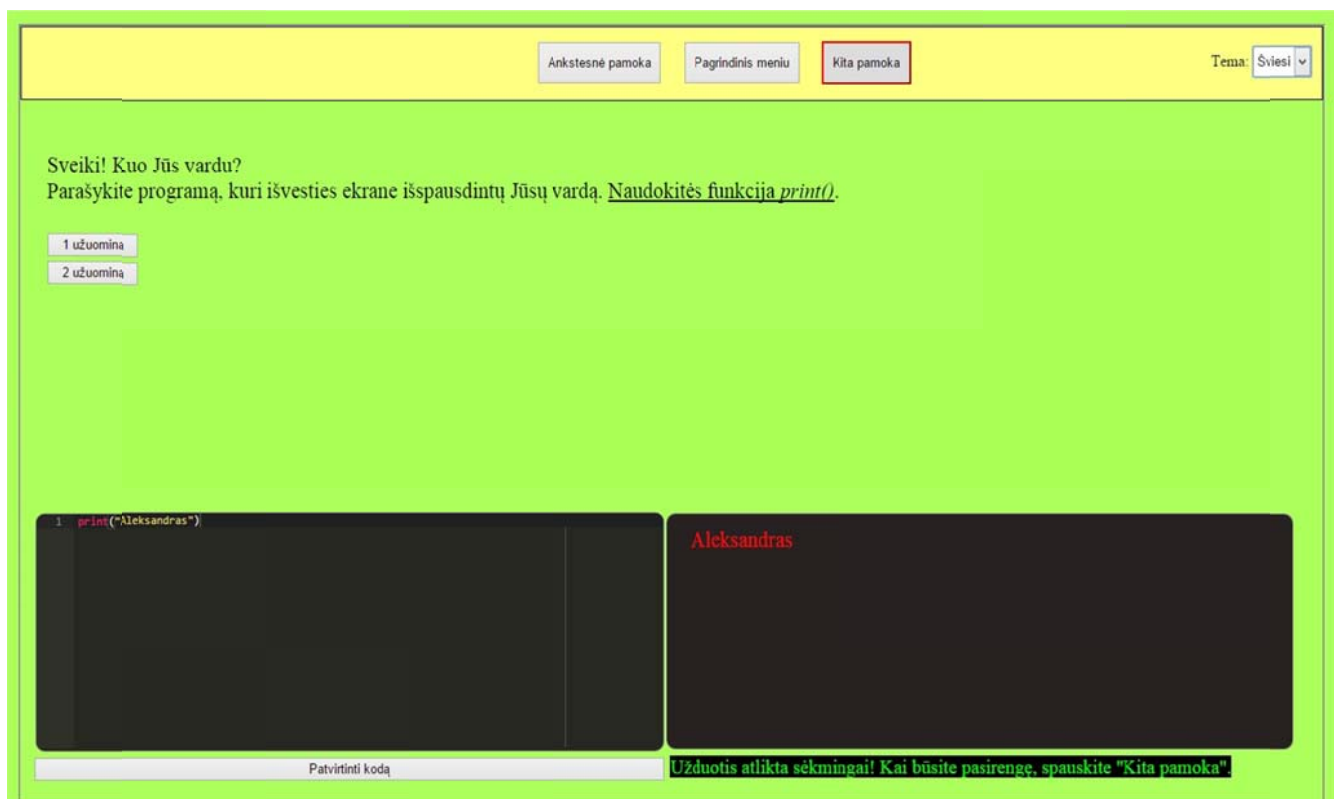
- *checkInput()*: Ar vartotojo pateiktame kode yra tam tikri būtini elementai užduočiai atlikti? Reikalaujame, kad pateiktyje būtų tam tikri *Python* kalbos elementai arba tam tikras tų elementų kiekis;
- *checkOutput()*: Ar vartotojo pateiktas kodas vykdomas grąžina reikiamas reikšmes? Reikalaujama, kad interpretatorius grąžintų simbolių eilutę, kuriai būdingas vienas ar keli požymiai (pavyzdžiui – simbolių eilutė turi susidėti vien iš skaitmenų).

Patenkinus pirmąjį reikalavimą kodas yra siunčiamas išoriniam kompiliatoriui, kuris grąžina tam tikrą išvestį simbolių eilutės pavidalu. Analizuojant šią simbolių eilutę konstatuojama, ar pateiktis tenkina antrąjį reikalavimą. Tik patenkinus abu šiuos reikalavimus vartotojo kodo pateiktis pripažįstama tinkama.

Duomenų analizė vykdoma naudojantis *Regular Expressions* metodais, kadangi įvedamų arba išvedamų duomenų atitikimo tikrinimas stipriai priklauso nuo konkrečios situacijos: kai kur duomenys turi būti griežtai tokie patys, kaip reikalaujama, o kai kur pakanka to, kad vartotojas panaudotų tam tikrą elementą, arba išvestis būtų reikiamo duomenų tipo. Dėl šios priežasties kiekvienai pamokai yra priskirtas atskiras *JavaScript* failas su minėtomis tikrinimo funkcijomis.

3.3.3. Grafinė vartotojo sąsaja

Vartotojo kodo įvedimui naudojama *Ace* biblioteka, sukurianti specializuotą kodo įvesties lauką, kuris ne tik leidžia patogią bei visavertę kodo įvestį (įskaitant tabuliaciją, spec. simbolius), tačiau taip pat dekoruoja įvedamą kodą – kiekvienos programavimo kalbos tipiniai sakiniai bei raktažodžiai atpažįstami bei išskiriami tam tikra spalva. Negana to, šis įrankis turi tam tikrų naudingų papildomų funkcijų, pavyzdžiui *getValue()*, kuri be papildomų *JavaScript* komandų pateikia įvestį simbolių eilutės pavidalu.

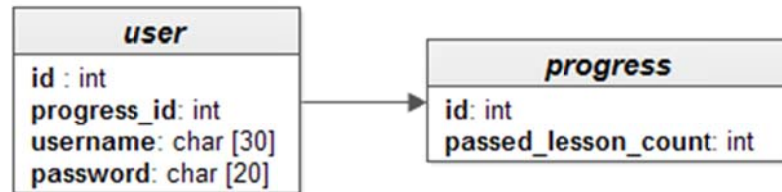


4 pav. Užduoties lango pavyzdys mokomojoje aplinkoje

Užduočių, meniu bei kodo išbandymo puslapiai kurti naudojantis *HTML* bei *CSS*, vartotojo sąsajos elementų sąveikai su vartotojo veiksmais aplinkoje realizuoti naudojamos *JavaScript* bei *jQuery* bibliotekos funkcijos.

3.3.4. Vartotojo duomenų bazė

Siekiant sudaryti vartotojui galimybę sekti savo progresą bei kontroliuoti vartotojui pasiekiamo turinio elementus buvo realizuota tokia duomenų bazė *py_aplinka*:



5 pav. Realizuotos duomenų bazės *py_aplinka* modelis.

Lentelėje *User* saugomi vartotojo prisijungimo duomenys; vartotojo vardo ilgis gali būti iki 30 simbolių, kai slaptažodis – iki 20. Laukelis *progress_id* leidžia susieti vartotojo lentelės įrašą su tam tikru *progress* lentelės įrašu, kur saugomas vartotojo progresas, t.y. eilės tvarka įveiktų užduočių kiekis.

3.3.5. Turinio prieinamumo valdymas

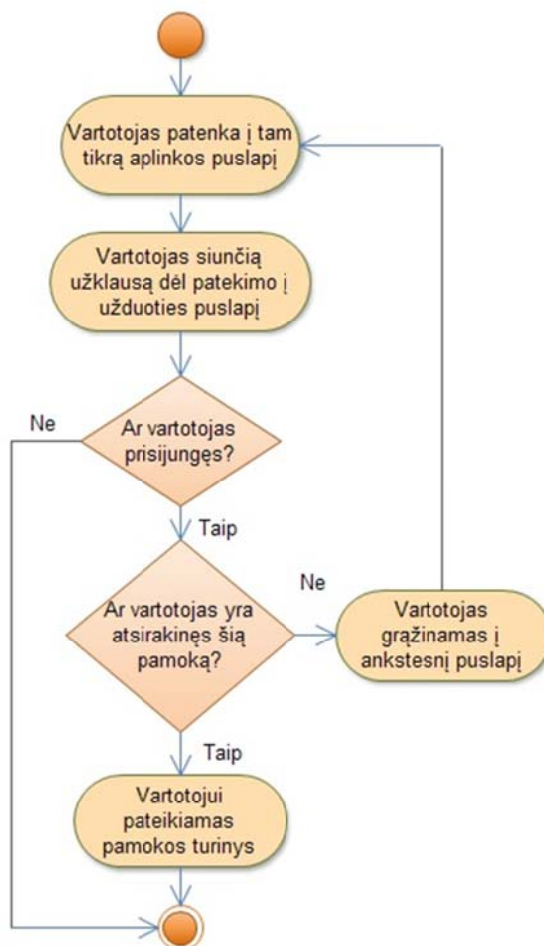
Sukurtos interaktyvios mokymosi aplinkos turinys prieinamas tik identifikuotiems vartotojams. Vartotojo paskyra kuriama registracijos puslapyje, kur įvedami duomenys siunčiami tikrinimui su duomenų bazėje esančiais paskyrų duomenimis siekiant išvengti dubliavimosi. Jeigu paskyra sukuriam sėkmingai, vartotojas gali su registracijos metu įvestais duomenimis prisijungti prie sistemos.

Įvedus teisingus prisijungimo duomenis siunčiama *AJAX* užklausa į php failą *login.php*, ten sukuriam nauja sesija, kurioje saugomas vartotojo prisijungimo vardas bei jo progresas (įveiktų užduočių aplinkoje kiekis). Vartotojas atsijungti iš aplinkos gali pagrindiniame meniu – tuomet siunčiama užklausa į failą *logout.php*, kuriame sunaikinami sesijos duomenys, o vartotojas nukreipiamas į prisijungimo puslapį. Tiesa, atsijungimas neužkerta kelio vartotojui vis vien patekti į tam tikrą aplinkos puslapį suvedus to puslapio adresą. Tam, kad išvengti tokių atvejų, atidarant kiekvieną aplinkos puslapį vykdoma *AJAX* užklausa į failą *logged_in.php*, kuriame nustatoma, ar vartotojas yra prisijungęs, kadangi turinys jam prieinamas tik tokiu atveju.

Mokymosi procesas vyksta vartotojui paeiliui atlikinėjant pateiktas užduotis. Matydamas sekančią užduotį vartotojas gali gauti užuominų apie ankstesnės užduoties atlikimą; iš kitos pusės, sekančių užduočių nepasiekiamumas labiau motyvuoja vartotoją įveikti paskutinę prieinamą. Dėl šių priežasčių prieš pateikiant vartotojui kiekvienos pamokos turinį, vykdoma *AJAX* užklausa į failą *isAllowed.php*, kuriame konkrečios užduoties numeris tikrinamas su iš php sesijos gaunamu įveiktų užduočių kiekiu (paskutinės įveiktos užduoties numeriu). Vartotojui leidžiama matyti visas įveiktas bei tolesnę užduotis.

Progresas atnaujinamas tik tuomet, kai sėkmingai įvykdoma anksčiau neįveikta užduotis. Tokiu atveju kreipiamasi į failą *updateProgress.php*, kuris atnaujina įveiktų pamokų skaičių tiek php sesijoje, tiek duomenų bazėje.

Pamokų turinio, pasiekiamo vartotojui, valdymą galima pavaizduoti tokia būsenų diagrama:



6 pav. Pamokų turinio pasiekiamumo valdymo diagrama

3.4. Darbo rezultatų analizė

Buvo suprojektuota ir sukurta interaktyvi mokymosi aplinka, skirta mokytis programuoti *Python* kalba, kurioje mokymosi procesas buvo paremtas praktinių programavimo žinių gavimu bei gilinimu. Į užduotis buvo integruotas nuotolinis kompiliatorius *Sphere Engine*, į kurį siunčiamas vartotojo kodas, priimamas naudojantis *Ace* kodo įvesties įrankiu, bei iš kompiliatoriaus gaunamas atsakas apie programos kodo vykdymo metu gautus duomenis (tekstinė išvestis, klaidos).

Realizuota duomenų bazė *py_aplinka*, skirta vartotojo paskyros bei progreso mokymosi procese saugojimui.

Aplinka pasiekama internete adresu: www.coderland.lt/ismok_python

3.5. Interaktyvios mokymosi aplinkos testavimas

Buvo testuojamas įvedamų duomenų tikrinimas registracijos puslapyje, *Sphere engine* kompiliatoriaus veikimas vartotojo kodo interpretavimo puslapyje bei pateikiamo kodo interpretavimas bei rezultatų vertinimas 2 pamokoje. Testavimas atliktas naudojantis šiais įrenginiais: asmeniniu kompiuteriu (*Windows 8.1*, *Opera* naršyklė) bei mobiliuoju telefonu *LG F70* (*Android 4.4.2*, *Chrome* naršyklė). Testavimo rezultatai pateikiami *priedų* skyriuje.

3.6. Patarimai, pastebėjimai, rekomendacijos

Tobulinant sukurta aplinką, reikėtų papildyti užduočių bei teorinės medžiagos kiekį, įtraukti žaidimizacijos elementų siekiant smagesnio bei malonesnio mokymosi proceso bei suteikti daugiau aplinkos personalizavimo galimybių vartotojui.

Sukurta 18 užduočių, skirtų suteikti vartotojui praktinių programavimo įgūdžių. Numatyta, kad sprendamas minėtas užduotis, vartotojas turėtų susipažinti su sąlyginių sakinių, ciklų, bei smulkių bazinių funkcijų (teksto išvedimo ekrane, komentavimo) ypatybėmis *Python* kalboje, taip pat – suprasti keletą paprastų algoritmų (minimalios/maksimalios reikšmės radimo, burbulo rikiavimo algoritmo) struktūrą bei prasmę.

Nors interaktyvios mokymosi aplinkos realizacijoje, esančioje *coderland.lt* svetainėje, mokomasis turinys pateikiamas tik *Python* programavimo kalbos mokymuisi, nėra techninių kliūčių realizuoti kursą, skirtą mokytis bet kurią kitą *Sphere Engine* kompiliatoriaus palaikomą programavimo kalbą.

4. IŠVADOS

- Remiantis atlikta populiarių virtualių mokymosi aplinkų apžvalga, nustatyta, kad visose jose būtinas dalyvis, turintis mokytojo/dėstytojo rolę. Nustatyta, jog daugumoje jų naudojami testai mokymosi progreso įvertinimui.
- Remiantis atlikta populiarių VMA apžvalga ir besimokančiųjų programuoti problemomis bei poreikiais, pastebimais programavimo tematikos forumuose (pavyzdžiui, *StackOverflow.com*), buvo nuspręsta, kad racionalu kurti interaktyvią *Python* kalbai mokytis skirtą aplinką, kuri būtų orientuota į praktinių programavimo žinių suteikimą besimokančiajam.
- Buvo sukurta interaktyvi mokymosi aplinka, kurioje nuotolinis kompiliatorius tikrina vartotojo kodo pateiktis bei pateikia atsaką, o vartotojo paskyros duomenys bei mokymosi proceso progresas saugomi duomenų bazėje.
- Buvo testuojamas įvedamų duomenų tikrinimas registracijos puslapyje, Sphere engine kompiliatoriaus veikimas vartotojo kodo vykdymo puslapyje bei pateikiamo kodo transliavimas ir rezultatų vertinimas vienoje iš pamokų. Iš testavimo rezultatų seka, jog sukurta interaktyvi mokymuisi programuoti *Python* kalba skirta aplinka veikia tinkamai.

LITERATŪRA IR INFORMACINIAI ŠALTINIAI

1. S.Daukilas, J.Kasperiūnienė, E. mokymosi kursų projektavimas ir realizavimas, Kaunas, 2011.
2. Angelo G. Kasperiūnienė J., Rutkauskienė D., Nuo didaktikos e. didaktikos link. E. mokymosi paradigmos, modeliai ir metodai. Technologija, Kaunas, 2010.
3. *Virtuali mokymosi aplinka* [interaktyvus]. [žiūrėta 2015-04-10]. Prieiga per internetą: <http://www.vipt.lt/cms/app?service=external/index&sp=2313&sp=2030>
4. *Kineo statistiniai duomenys apie e. mokymąsi* [interaktyvus]. [žiūrėta 2015-04-10]. Prieiga per internetą: <http://elearningindustry.com/kineo-e-learning-in-the-enterprise-survey-results-2013-infographic>
5. *Surrey universiteto student pasitenkinimo e. mokymusi kitimas*. [interaktyvus] [žiūrėta 2015-04-15]. Prieiga per internetą: <http://www.d2l.com/press/releases/desire2learn-university-surrey-unveil-compelling-data-increased-student-satisfaction-virtual-learning/>
6. *Populiarių mokymosi aplinkų palyginimas*. [interaktyvus] [žiūrėta 2015-04-15]. Prieiga per internetą: <http://elearningindustry.com/lms-accessibility-comparison-blackboard-vs-desire2learn-vs-moodle-vs-sakai>.
7. *Moodle*. [interaktyvus] [žiūrėta 2015-04-15]. Prieiga per internetą: <http://en.wikipedia.org/wiki/Moodle>.
8. A. Rudaitis, Nuotolinių studijų kurso “Programavimas Python kalba“ projektavimas MOODLE sistema, 2011.
9. *SAKAI project*. [interaktyvus] [žiūrėta 2015-04-16]. Prieiga per internetą: http://en.wikipedia.org/wiki/Sakai_Project.
10. R. Mačiulskis, S.Turskienė (2011), Struktūrizuotos mokomosios medžiagos pateikimo virtualioje mokymosi aplinkoje Sakai galimybių analizė, XV kompiuterininkų konferencijos mokslo darbai, Klaipėda, Žara, p. 108- 117.
11. S.Turskienė, R.Mačiulskis, (2010), Nuotolinio mokymosi aplinkų SAKAI ir MOODLE galimybių lyginamoji analizė, Informacinių technologijų taikymas švietimo sistemoje 2010: e-studijų patirtis, aktualijos ir perspektyvos, Straipsnių rinkinys, ISSN 1822- 7244, Kauno kolegija, Kaunas, p. 62-66
12. *WebCT*. [interaktyvus] [žiūrėta 2015-04-16]. Prieiga per internetą: <http://en.wikipedia.org/wiki/WebCT>.

13. *FirstClass*. [interaktyvus] [žiūrėta 2015-04-16]. Prieiga per internetą: <http://en.wikipedia.org/wiki/FirstClass>.
14. *Brightspace*. [interaktyvus] [žiūrėta 2015-04-16]. Prieiga per internetą: <http://betakit.com/desire2learn-launches-new-integrated-learning-platform-called-brightspace>.
15. *CodeAcademy informacinis puslapis*. [interaktyvus] [žiūrėta 2015-04-10]. Prieiga per internetą: <http://www.codecademy.com/about>.
16. *MAIK Žaidimų programavimas su Python*. [interaktyvus] [žiūrėta 2015-05-18]. Prieiga per internetą: <http://vivu.lt/course/view.php?id=98>.
17. *Hour of Python by Trinket*. [interaktyvus] [žiūrėta 2015-05-20]. Prieiga per internetą: <https://hourofpython.com/>.
18. *Coderland.lt forumo skiltis*. [interaktyvus] [žiūrėta 2015-05-10]. Prieiga per internetą: <http://coderland.lt/programavimo-forumas/>.
19. *Stackoverflow diskusijų forumas*. [interaktyvus] [žiūrėta 2015-05-10]. Prieiga per internetą: <http://stackoverflow.com/questions/>.
20. V.Šarkys, Python programavimo kalbos plėtinių kūrimo galimybių analizė, kursinis darbas, Šiaulių universitetas, Šiauliai, 2014.
21. *TIOBE programavimo kalbų populiarumo indeksas*. [interaktyvus] [žiūrėta 2015-04-15]. Prieiga per internetą: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html/>.
22. J.Kairiūkštis, Internetinio kompiliatoriaus/interpretatoriaus pritaikymas programavimo mokymosi portale, Šiauliai, 2013.
23. *Application programming interface*. [interaktyvus] [žiūrėta 2015-03-17]. Prieiga per internetą: http://en.wikipedia.org/wiki/Application_programming_interface.
24. *Coding ground kompiliatorius*. [interaktyvus] [žiūrėta 2015-04-20]. Prieiga per internetą: <http://www.tutorialspoint.com/codingground.htm>.
25. *Codepad kompiliatorius*. [interaktyvus] [žiūrėta 2015-04-20]. Prieiga per internetą: <http://codepad.org/about>.
26. *Repl.it interpretatorius*. [interaktyvus] [žiūrėta 2015-03-17]. Prieiga per internetą: <http://repl.it>.
27. *Trinket.io įrankis*. [interaktyvus] [žiūrėta 2015-05-18]. Prieiga per internetą: <https://ideone.com/sphere-engine>.
28. *Sphere Engine kompiliatorius*. [interaktyvus] [žiūrėta 2015-04-17]. Prieiga per internetą: <https://ideone.com/sphere-engine>.

PRIEDAI

Interaktyvios aplinkos testavimas

1. Registracijos lange įvedamų duomenų interpretavimo testavimas.

1.1. Įvedami nesutampantys slaptažodžiai.

Įvestis:

Vartotojo vardas: Vartotojas

Slaptažodis: Slaptasis

Pakartoti: Slaptasi5

Rezultatas:

Slaptažodžiai nesutampa!.

1.2. Įvedamas jau egzistuojančios vartotojo paskyros slaptyvardis.

Įvestis:

Vartotojo vardas: Vartotojas

Slaptažodis: labaislaptas

Pakartoti: labaislaptas

Rezultatas:

Toks vartotojas jau yra!

1.3. Įvedami tinkami registracijos duomenys.

Įvestis:

Vartotojo vardas: Studentas

Slaptažodis: noriuvalgyt

Pakartoti: noriuvalgyt

Rezultatas:

Sėkmingai užsiregistravote! Dabar galite prisijungti.

2. Kodo išbandymo lango testavimas.

2.1. Įvedamas minimalios *Python* programos kodas.

Įvestis:

```
for i in range(1, 5):  
    print(i)
```

Rezultatas:

1 2 3 4

3. Užduoties atlikimas 2 pamokoje (užduotis: programiškai apskaičiuoti ir išvesti ekrane savo vardo ilgį, naudojant *Python* funkciją *len()*).

3.1. Skaičiuojamas vardo ilgis be atsakymo išvedimo:

Ivestis:

```
len("Vidmantas")
```

Rezultatas:

Kažkas ne taip... Patikrinkite kodą ir bandykite dar kartą.

3.2. Ekrane išvesties pateikiama daugiau, negu prašoma užduotyje:

Ivestis:

```
print(len("Vidmantas"), " simboliai")
```

Rezultatas:

9 simboliai

Kažkas ne taip... Patikrinkite kodą ir bandykite dar kartą.

3.3. Pagal reikalavimus atliekama užduotis.

Ivestis:

```
print(len("Vidmantas"))
```

Rezultatas:

9

Užduotis atlikta sėkmingai! Kai būsite pasirengę, spauskite "Kita pamoka".

Konferencijų programos ir sertifikatai

KRETINGOS SUAUGUSIŲ IR JAUNIMO MOKYMO CENTRAS
II RESPUBLIKINĖ MOKSLINĖ – PRAKTINĖ KONFERENCIJA
„ŠIUOLAIKINIŲ INFORMACINIŲ IR RYŠIŲ TECHNOLOGIJŲ PANAUDOJIMO
ĮTAKA UGDYMO KOKYBEI“

2015 m. balandžio 30 d. 10.00 val.

Vilj Pranciškonių gimnazijos Šv. Antano rūmai, Vilniaus g. 6, Kretinga

- 9⁰⁰ – 10⁰⁰ Konferencijos dalyvių registracija
 10⁰⁰ – 10³⁰ Konferencijos dalyvių pasveikinimas

Plenarinis posėdis. Pirmininkė doc. dr. Sigita Turskienė

- 10³⁰ – 11⁰⁰ *IEA Tarptautinis kompiuterinio ir informacinio raštingumo tyrimas ICILS 2013: rezultatai ir pamokos*
 Asta Buinevičiūtė, Švietimo informacinių technologijų centras
 11⁰⁰ – 11³⁰ *IKT projektai mokyklai*
 Alvida Lozdienė, Ugdymo plėtotės centras
 11³⁰ – 12⁰⁰ *Atviras profesinis bendradarbiavimas*
 Dr. Estela Daukšienė, Lietuvos nuotolinio ir e. mokymosi (LieDM) asociacija
 12⁰⁰ – 12³⁰ *Svarbiausia informatikos mokslo problema*
 Prof. habil. dr. Leonidas Sakalauskas, Šiaulių universitetas
 12³⁰ – 13³⁰ **Pietų pertrauka**

Konferenciją galėsite stebėti internetu adresu distance.su.lt/kretinga/

2 posėdis. Nuotolinis mokymas(is) (I sekcija)

Pirmininkas doc. dr. Rytis Vilkonis

Vilj Pranciškonių gimnazijos Šv. Antano rūmai, Vilniaus g. 6, Kretinga

- 13³⁰ – 13⁴⁵
Interaktyvūs mokymo formų ir metodų taikymas lietuvių kalbos ir literatūros pamokose
 Lina Sirgėdienė, Vilniaus suaugusiųjų mokymo centras
 13⁴⁵ – 14⁰⁰
Mokymo ir mokymosi praktika suaugusiųjų nuotoliniame mokyme(is)
 Palmira Jurgilienė, Kretingos suaugusiųjų ir jaunimo mokymo centras

- 14⁰⁰ – 14¹⁵
Kompiuterinės lingvistikos galimybės ir praktika gimtosios kalbos ugdymo procese
 Audronė Anulienė, Tauragės suaugusiųjų mokymo centras

- 14¹⁵ – 14³⁰
Inovacijų kūrimo projektai: papildytosios realybės technologijos pritaikymo švietime galimybių paieška
 Doc. dr. Rytis Vilkonis, Šiaulių universitetas
 14³⁰ – 14⁴⁵
Virtuali ekskursija – efektyvus suaugusiųjų mokymo(is) metodas užsienio kalbos pamokose
 Eglė Morozova, Klaipėdos suaugusiųjų gimnazija

- 15⁰⁰ – 15¹⁵
Mūraus modelio įgyvendinimas suaugusiųjų švietime
 Vilma Račaitienė, Kauno suaugusiųjų mokymo centras

- 15¹⁵ – 15³⁰
Virtualiųjų įrankių panaudojimas e. mokymui
 Dr. Liudvikas Kaldauskas, Šiaulių universitetas
 15³⁰ – 15⁴⁵
IKT ir savarankiškas mokymasis
 Dr. Rasa Pocevičienė, Šiaulių universitetas
 Doc. dr. Sigita Turskienė, Šiaulių universitetas
 Vidmantas Šarkys, Šiaulių universitetas
 15⁴⁵ – 16⁰⁰

2 posėdis. IT ir ryšių panaudojimo įtaka ugdymo kokybei

(II sekcija)

Pirmininkė dr. Vaida Bartkutė – Norkūnienė

Kretingos suaugusiųjų ir jaunimo mokymo centras, J. Pabrėžos g. 8, Kretinga

- 13³⁰ – 13⁴⁵
IT naudojimas integruotoje pamokoje, konkurse – iššūkis mokymui ir mokymuisi
 Nina Matuzienė, Klaipėdos „Ažuolyno“ gimnazija
 13⁴⁵ – 14⁰⁰
Švietimas skaitmeniniame amžiuje. Mokymosi veiksmingumas ugdant XXI amžiaus įgūdžius
 Anton Petrovskij, Klaipėdos Gedminų progimnazija, Vilj „Edukateka“
 Gediminas Jonauskas, Klaipėdos Gedminų progimnazija, Vilj „Edukateka“
 Saulius Rudelis, Klaipėdos Gedminų progimnazija, Vilj „Edukateka“
 14⁰⁰ – 14¹⁵
Projektų metodas kaip galimybė ugdyti mokinio kompetencijas
 Vilma Rekešienė, Šiaulių Julius Janonio gimnazija
 Daiva Riukienė, Šiaulių Julius Janonio gimnazija
 Andrius Bukis, Šiaulių Simono Daukanto gimnazija
 14¹⁵ – 14³⁰
Projektiniai integruoti matematikos – IT darbai
 Margiris Burakauskas, Šiaulių Julius Janonio gimnazija
 Džiugas Mažulis, Šiaulių Julius Janonio gimnazija
 14³⁰ – 14⁴⁵
Projektiniai integruoti matematikos – IT darbai
 Lukas Martišius, Šiaulių Julius Janonio gimnazija

14⁴⁵ – 15⁰⁰

Kavos pertrauka

- 15⁰⁰ – 15¹⁵
Mokymosi motyvacijos didinimas naudojant patiekimų vertinimą virtualioje mokymosi aplinkoje
 Leonas Simkus, Klaipėdos „Vėtrungės“ gimnazija
 15¹⁵ – 15³⁰
Mokinio įtraukimo į aktyvų mokymąsi galimybės
 Dalia Balčiūnienė, Kelmės rajono Kražių gimnazija
 15³⁰ – 15⁴⁵
Medijų ir netradicinių erdvių panaudojimas aktualizuojant etninę kultūrą lietuvių kalbos pamokose
 Ramunė Galdikienė, Klaipėdos „Aukuro“ gimnazija

- 15⁴⁵ – 16⁰⁰
Logika informatikoje: nuo Aristotelio iki kompiuterinės logikos
 Doc. dr. Vaida Bartkutė – Norkūnienė, Utenos kolegija
 Tadas Panavas, Utenos kolegija

16⁰⁰ – 16³⁰

Diskusijos. Konferencijos uždarymas

Vilj Pranciškonių gimnazijos Šv. Antano rūmai, Vilniaus g. 6, Kretinga



KRETINGOS RAJONO PEDAGOGŲ ŠVIETIMO CENTRAS

Savivaldybės biudžetinė įstaiga, J.Pabrėžos g. 8, 97129 Kretinga, tel. (8 445) 51363 / 79192,
el. p. pedagogusc@psckretingos.lt, <http://www.psckretingos.lt>.
Duomenys saugomi ir kaupiami Juridinių asmenų registre, kodas 195175933

Dalyviams pagal sąrašą

PAŽYMA DĖL METODINĖS VEIKLOS VYKDYMO

2015-04-30 K3-M Nr. 94
Kretinga

Pažymima, kad 2015-04-30 sąraše nurodyti asmenys organizavo respublikinę mokslinę-praktinę konferenciją „Šiuolaikinių informacinių ir ryšių technologijų panaudojimo įtaka ugdymo kokybei“.

PRIDEDAMA. Dalyvių sąrašas, 2 lapai.

Direktorė



Adelė Mazeliauskienė

Respublikinė mokslinė-praktinė konferencija
„Šiuolaikinių informacinių ir ryšių technologijų panaudojimo įtaka ugdymo kokybei“
 2015-04-30

SĄRAŠAS

Eil. Nr.	Vardas, pavardė	Institucija, darbovietė	Veikla
1.	Palmira Jurgilienė	Kretingos suaugusiųjų ir jaunimo mokymo centras	lektorė, organizatorė
2.	Antanas Sungaila	Kretingos rajono savivaldybės administracijos švietimo skyrius	organizatorius
3.	Adelė Mazeliauskienė	Kretingos rajono pedagogų švietimo centras	organizatorė
4.	Irena Dukštienė	Kretingos suaugusiųjų ir jaunimo mokymo centras	organizatorė
5.	Aleksas Narščius	Klaipėdos universitetas	organizatorius
6.	Tomas Bersėnas	Šiaulių universitetas	organizatorius
7.	Airina Volungevičienė	LieDM asociacija	organizatorė
8.	Natalija Kimso	Vilniaus suaugusiųjų mokymo centras, LSMCVA	organizatorė
9.	Sigita Turskienė	Šiaulių universitetas	lektorė, organizatorė
10.	Asta Buinevičiūtė	Švietimo informacinių technologijų centras	lektorė
11.	Alvida Lozdienė	Ugdymo plėtotės centras	lektorė
12.	Estela Daukšienė	Lietuvos nuotolinio ir e. mokymosi (LieDM) asociacija	lektorė
13.	Leonidas Sakalauskas	Šiaulių universitetas	lektorius
14.	Rytis Vilkonis	Šiaulių universitetas	lektorius, organizatorius
15.	Lina Sirgedienė	Vilniaus suaugusiųjų mokymo centras	lektorė
16.	Audronė Anulienė	Tauragės suaugusiųjų mokymo centras	lektorė
17.	Eglė Morozova	Klaipėdos suaugusiųjų gimnazija	lektorė
18.	Vilma Račaitienė	Kauno suaugusiųjų mokymo centras	lektorė
19.	Liudvikas Kaklauskas	Šiaulių universitetas	lektorius, organizatorius
20.	Rasa Pocevičienė	Šiaulių universitetas	lektorė
21.	Sigita Turskienė	Šiaulių universitetas	lektorė
22.	Vidmantas Šarkys	Šiaulių universitetas	lektorius
23.	Vaida Bartkutė – Norkūnienė	Utenos kolegija	lektorė
24.	Nina Matuzienė	Klaipėdos „Ažuolyno“ gimnazija	lektorė
25.	Anton Petrovskij	Klaipėdos Gedminų progimnazija, VšĮ „Edukatėka“	lektorius
26.	Gediminas Jonauskas	Klaipėdos Gedminų progimnazija, VšĮ „Edukatėka“	lektorius
27.	Saulius Rudelis	Klaipėdos Gedminų progimnazija, VšĮ „Edukatėka“	lektorius
28.	Vilma Rekešienė	Šiaulių Juliaus Janonio gimnazija	lektorė
29.	Daiva Riukienė	Šiaulių Juliaus Janonio gimnazija	lektorė
30.	Andrius Bukis	Šiaulių Simono Daukanto gimnazija	lektorius
31.	Margiris Burakauskas	Šiaulių Juliaus Janonio gimnazija	lektorius
32.	Džiugas Mažulis	Šiaulių Juliaus Janonio gimnazija	lektorius

STUDENTŲ MOKSLINĖ PRAKTINĖ KONFERENCIJA

MOKSLAS IR STUDIJS 2015: TEORIJA IR PRAKTIKA

2015 gegužės 14 d. Šiaurės Lietuvos kolegija (Tilžės g. 22, Šiauliai)

KONFERENCIJOS PROGRAMA

9.00-10.00	Dalyvių registracija	
PLENARINIS POSĖDIS (405 a.) Moderatorė – Ieva Bilbokaitė , Šiaurės Lietuvos kolegijos mokslinės veiklos koordinatore		
10.00-10.05	<i>Sveikinimo žodis konferencijos dalyviams</i>	Živilė Nakčiūnienė Šiaurės Lietuvos kolegijos Direktoriaus pavaduotoja akademinėi veiklai
10.05-10.10	<i>Supažindinimas su konferencijos darbotvarke</i>	Ieva Bilbokaitė Šiaurės Lietuvos kolegijos Mokslinės veiklos koordinatore
10.10-10.30	<i>Interaktyvi aplinka, skirta Python programavimo kalbos mokymuisi</i>	Vidmantas Šarkys Šiaulių universiteto studentas
10.30-10.50	<i>Fizinių ir juridinių asmenų bankroto proceso lyginamoji analizė</i>	Deimantė Tamošauskienė Šiaurės Lietuvos kolegijos studentė
10.50-11.10	<i>Vadybinių kompetencijų plėtojimas verslo praktinio mokymo firmoje</i>	Judita Putriūtė Šiaurės Lietuvos kolegijos asistentė, Viktorija Tamulytė Šiaurės Lietuvos kolegijos studentė
11.10	DARBAS SEKCIJOSE, kava, užkandžiai	

PAŽYMĖJIMAS

VIDMANTAS ŠARKYS

2015 m. gegužės 14 d.

dalyvavo nacionalinėje studentų mokslinėje praktinėje konferencijoje

„**Mokslas ir studijos 2015: teorija ir praktika**“

skaitė pranešimą plenariniame posėdyje

„Interaktyvi aplinka, skirta Python programavimo kalbos mokymuisi“.

Šiaurės Lietuvos kolegijos

Direktorius

Dr. Mykolas Dromantas



2015 m. gegužės 14 d.
Registracijos Nr. P3-2515

ANOTACIJA

Darbe apžvelgtos populiarios virtualios mokymosi aplinkos. Remiantis pagrindiniais besimokančiųjų programuoti asmenų poreikiais suprojektuota interaktyvi mokymosi aplinka, orientuota į praktinių žinių suteikimą. Pagal projektą sukurta interaktyvi mokytis programuoti *Python* kalba skirta aplinka, naudojanti išorinį kompiliatorių bei duomenų bazę vartotojo paskyros duomenims saugoti.

SUMMARY

In this paper popular virtual learning environments are reviewed. By having the main needs of those who study programming in mind, interactive learning environment, oriented towards practical programming experience, was designed. Using this design, interactive learning environment for learning *Python*, uses online compiler and database to store user's account information, was created.