

ŠIAULIŲ UNIVERSITETAS

Informacinių technologijų katedra

Denas Pavlavičius

**Klasių diagramos generavimas iš Java kalbos
šaltinio tekstų**

Bakalauro baigiamasis darbas

Vadovė asist. S. Ramanauskaitė

Šiauliai, 2010

DARBO SANTRAUKA

Tikslas	Sukurti programą, kuri iš pateiktų Java projekto šaltinio kodo failų sugeneruotų klasių diagramą.
Analitinėje dalyje nagrinėjami aspektai	Klasių diagramos notacijos analizavimas. Panašios paskirties programų palyginimas.
Kuriamo produkto prototipai	Microsoft Office Visio – didelės apimties programa, skirta visokioms, tame tarpe ir UML, diagramoms braižyti. „AutoDiagrammer“ – programa skirta klasių diagramų braižymui iš parašyto programos kodo. „ClassDraw“ – klasių diagramų braižymo programa pagrįsta „vedlių“ principu. „Eclipse“ – nemokama programa skirta programuotojams. Tai programų kūrimo įrankis, tačiau turintis galimybę iš parašyto programos kodo nubraižyti klasių diagramą. „NetBeans“ – produktas, skirtas programų kūrimui ir turintis papildomą modulį klasių diagramoms braižyti ir generuoti iš šaltinio kodo.
Technologinis sprendimas	Java programavimo kalba
Panaudota programinė įranga	NetBeans IDE 6.7.1 – Java kalbos kodo rašymui bei kompiliavimui; Microsoft Office Visio – diagramų braižymui; Microsoft Word – darbo ataskaitos rengimui.
Naudotos UML diagramos	Veiklos konteksto diagrama; Panaudos atvejų diagrama; Duomenų modelis; Duomenų surinkimo veiklos diagrama; Diagramos generavimo sekų diagrama; Programos būsenų diagrama; Programos objektų bendradarbiavimo diagrama; Programos duomenų struktūrų diagrama; Programos komponentų diagrama.
Testavimo apimtis	Funkcinis testavimas; Našumo testavimas; Atstatymo testavimas.
Vartotojo dokumentacijos sudėtinės dalys	Sistemos funkcinis aprašymas; Sistemos vartotojo vadovas; Diegimo instrukcijos.

SUMMARY

Generation of Class Diagram from Java Source Code

The Class diagram clearly presents program's structure and reciprocal interaction of objects. In order to create class diagram requires a lot of time and thoroughness. Such diagrams should be created before starting to write program's code because this diagram helps to orient which objects to create and how to connect them. Usually, the succession of works is opposite – firstly, program's code is written and only then class diagram is created.

In this project the main notation of class diagram is reviewed. In addition, solutions, that already exist and are devoted to create classes' diagrams or to generate from source's code are compared. Based on that, requirements and architecture specifications are described. The programme is created on the basis of those specifications. Also, a particular testing was executed for the created programme. The program's testing itself and the guidelines for using it are described in details.

In conclusion, the created programme meets all the needed requirements and is able to generate class diagrams automatically from Java sources' codes, provides a possibility to edit diagram by adding something or making some changes. Also, it allows to model new classes and their relationship with other classes that already exist.

TURINYS

IVADAS.....	6
1. KLASIŲ DIAGRAMA IR JOS GENERAVIMO PRIEMONĖS.....	7
1.1 Kласių diagrama ir jos notacija.....	7
1.2 Kласių diagramos generavimo galimybės/programos	10
1.3 Programų palyginimas	13
2. REIKALAVIMŲ SPECIFIKACIJA.....	15
2.1 Programinės įrangos paskirtis	15
2.2 Projekto dalyviai	15
2.3 Vartotojai	15
2.4 Įpareigojantys apribojimai	15
2.5 Veiklos kontekstas	16
2.6 Produkto veiklos sfera.....	16
2.7 Funkciniai reikalavimai.....	19
2.8 Nefunkciniai reikalavimai.....	21
2.9 Atviri klausimai.....	22
2.10 Egzistuojantys sprendimai	22
2.11 Pritaikymas.....	23
2.12 Galimos sistemos kūrimo rizikos	23
2.13 Vartotojo dokumentacija ir apmokymas	23
2.14 Perspektyviniai reikalavimai.....	23
3. PROGRAMINĖS ĮRANGOS ARCHITEKTŪROS SPECIFIKACIJA	24
3.1 Architektūros specifikacijos paskirtis	24
3.2 Architektūros pateikimas	24
3.3 Architektūros tikslai ir apribojimai	24
3.4 Sistemos dinaminis vaizdas.....	24
3.5 Sistemos statinis vaizdas.....	27
3.6 Programinės įrangos komponentai	31
3.7 Programinės įrangos kokybė.....	31
4. TESTAVIMAS	32
4.1 Testavimo paskirtis	32
4.2 Programinės įrangos funkcijų testavimas.....	32
4.3 Programinės įrangos charakteristikų testavimas	35
4.4 Testavimo išvados.....	37
5. VARTOTOJO DOKUMENTACIJA.....	38
5.1 Vartotojo vadovo paskirtis	38
5.2 Sistemos funkcinis aprašymas.....	38
5.3 Sistemos vartotojo vadovas.....	38
5.4 Sistemos diegimo vadovas	44
IŠVADOS	45
LITERATŪRA.....	46
TERMINŲ IR SANTRUMPŲ ŽODYNĖLIS.....	47

PRIEDAI	48
1 priedas. Klasių diagramų braižymo įrankių palyginimas	48
2 priedas. Klasių diagramų braižymo testavimas	50

IVADAS

Klasių diagrama pateikia aiškų programos struktūros ir tarpusavio sąveikos vaizdą, tačiau jos sudarymas gali pareikalauti daug kruopštumo ir laiko. Paprastai šios diagramos turėtų būti braižomos prieš pradedant programos kodo rašymą, nes pagal šią diagramą galima orientuotis kokius objektus kurti ir kaip juos tarpusavyje susieti, bet dažniausiai šių darbų eiliškumas būna atvirkštinis – pirma parašomas programos kodas ir tik tada braižoma klasių diagrama.

Šio darbo tikslas – sukurti programą, kuri iš pateiktų Java projekto šaltinio kodo failų sugeneruotų klasių diagramą. Tokiu būdu būtų sumažintas žmogiškojo faktoriaus įsikišimas ir palengvintas klasių diagramos braižymas.

Darbo uždaviniai:

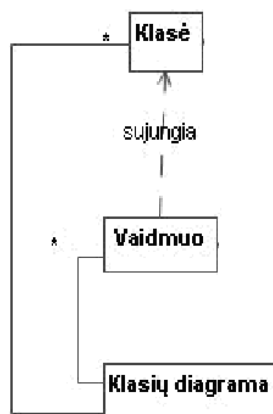
- Susipažinti su klasių diagramos notacija.
- Parengti reikalavimų bei architektūros specifikacijas.
- Realizuoti klasių diagramų braižymo programą ir atlikti išsamų jos testavimą.
- Realizuotai programai pateikti vartotojo vadovą.

1. KLASIŲ DIAGRAMA IR JOS GENERAVIMO PRIEMONĖS

Objektiškai orientuotų programų dokumentų modeliavimui ir specifikacijų kūrimui yra sukurta speciali kalba – UML (*ang. Unified Modeling Language*, Vieninga modeliavimo kalba). Ji padeda orientuotis programos kūrime tiek programuotojui, tiek užsakovui, nes yra pakankamai lengvai suprantama. [9]

1.1 Klasių diagrama ir jos notacija

Viena iš šios modeliavimo kalbos dalių yra klasių diagrama. Klasių diagrama atspindi objektiškai orientuotos programos struktūrą ir objektų tarpusavio ryšius. Tokios diagramos metamodelis pateiktas 1 pav. Pagrindiniai elementai – klasės ir asociacijos tarp jų. Klasė apibūdina objektų aibę, o asociacija apibūdina ryšių aibę.[7]



1 pav. Supaprastintas UML klasių diagramos modelis

1.1.1 Klasės elemento struktūra.

Klasės elementas vaizduojamas stačiakampio formos, kuris yra suskirstytas į tris dalis. Pirmoje dalyje (nuo viršaus) užrašomas klasės pavadinimas. Antroje surašomi klasės atributai – savybės. Trečioje dalyje aprašomi klasės metodai – veiksmai (žr. 2 pav.).

Klasės vardas
Atributo vardas: tipas = pradinė reikšmė
Operacijos vardas()

2 pav. Klasės struktūra

Klasės atributų bei metodų aprašymas vyksta pagal nustatytą klasių diagramos notaciją:

- Klasės atributai

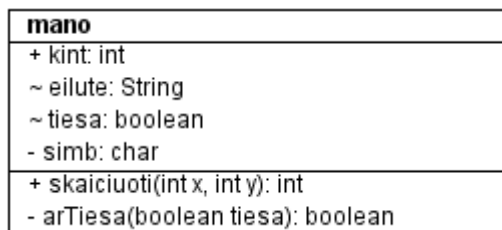
Pirmas ženklas prieš atributo pavadinimą yra nurodantis atributo matomumą, t.y.:

- „public“ atributas žymimas „+“ ženklu
- „private“ atributas žymimas „-“, ženklu
- „package“ atributas žymimas „~“ ženklu

Po matomumo ženklo seka atributo pavadinimas, dvitaškio ženklas, atributo tipas ir pradinė reikšmė (jeigu tokia nurodyta).

- Klasės metodai

Metodų kaip ir atributų aprašymas prasideda nuo matomumo ženklo, kurie yra tokie pat kaip ir atributų. Po jo seka metodo pavadinimas, skliaustuose perduodami parametrai su nurodytais tipais ir po skliaustų grąžinama reikšmė, jeigu metodas negrąžina jokios reikšmės tuomet nieko nerašoma [2, 7].

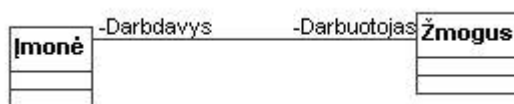


3 pav. Klasės bloko pavyzdys

1. 1. 2 Ryšiai tarp klasių

Pagrindiniai ryšiai tarp klasių yra:

- Asociacija – specifikuoja dvipusį ryšį tarp klasių. Asociacijos pabaiga yra vadinama vaidmeniu. Vaidmuo gali turėti vardą. 4 pav. pateiktas asociacijos vaidmenų pavyzdys: žmogus įmonėje užima darbuotojo vaidmenį, o įmonė vaidina darbdavio vaidmenį.



4 pav. Asociacijos pavyzdys

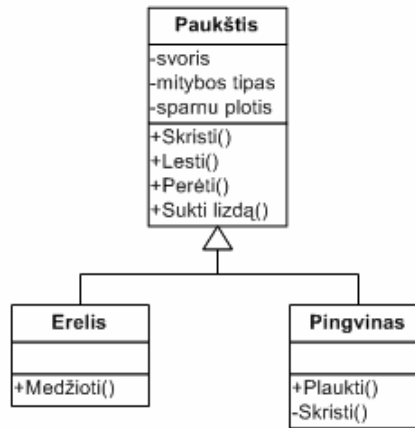
Asociacija turi mažiausiai du vaidmenis, kurie apibūdina, kokį vaidmenį užima klasė kitos klasės atžvilgiu (žr. 4 pav.).

Kiekvienas asociacijos vaidmuo turi daugybiškumo reikšmę, kuri parodo kiek vienos klasės objektų, gali būti susieta su kitos klasės objektais. 5 pav. pateiktas daugybiškumo naudojimo pavyzdys: žmogus gali turėti nulį ar daugiau sąskaitų (0..*), o sąskaita turi priklausyti bent vienam žmogui [2].



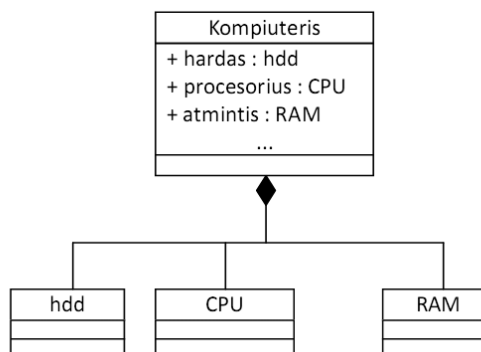
5 pav. Daugybiškumo pavyzdys

- Generalizacija (arba paveldėjimas) parodo, kad viena klasė (vaikas) yra susijusi su kita klase (tėvu) tokiu būdu, kad pirmąją galima naudoti visur, kur tinka antroji. Šis ryšys naudojamas tada, kai viena klasė dalijasi savo savybėmis su kitomis klasėmis. Paveldėjimo atveju vaiko klasės paveldi tėvo klasės duomenis ir metodus. Tačiau viena klasė negali paveldėti dviejų ar daugiau klasių duomenų ar metodų, o tik vienos. Klasių diagramoje paveldėjimo ryšys vaizduojamas rodykle nuo vaiko klasės į tėvo klasę, kur ties tėvo klase yra tuščia vidurinė rodyklė.[2, 7]



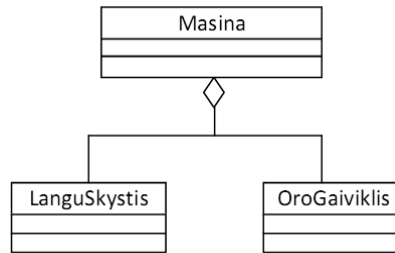
6 pav. Generalizacijos pavyzdys

- Priklausomybė – vienkryptis ryšys, nurodantis, kad vienas elementas naudoja kitą elementą. Priklausomybė skirstoma į du potipius:
 - Kompozicija – naudojama tada, kai vienas objektas susideda iš kitų objektų. Kompozicija pasižymi tuo, kad tik sukuriant objektą, kartu sukuriami ir jį sudarantys objektai, o naikinant pirmąjį, jį sudarantys objektai taip pat naikinami kartu (t.y. negali būti sunaikinti anksčiau). Trumpai tariant, pagrindinis objektas yra sukuriamas ir sunaikinamas vienu metu kaip ir visas sudėtinis objektas. Tai tarsi parodo, kad pagrindinis objektas negali egzistuoti be jį sudarančių objektų. Klasių diagramoje kompozicijos ryšys vaizduojamas, kaip rodyklė nuo „mažųjų“ objektų, į tą, kurį jie sudaro, tik vietoj rodyklės, ties pagrindiniu objektu yra vaizduojamas pilnaviduris (juodai užtušuotas) rombas (žr. 7 pav.). [2, 7]



7 pav. Kompozicijos ryšio pavyzdys

- Agregacija – labai panašus ryšys į kompoziciją. Čia taip pat vienas objektas susideda iš kelių kitų, bet pagrindinis skirtumas tas, kad čia sudedamieji objektai ne visada privalomi, t.y. gali būti situacijų, kai pagrindiniam objektui visiškai neprireiks sukurti jį sudarančio objekto, ar tas objektas bus sukurtas ir greitai sunaikintas ir pan. Kadangi tai labai panašu į kompoziciją, tai ir klasių diagramoje žymima panašiai – nuo objektų, sudarančių kažkokį tai kitą objektą piešiama rodyklė link pagrindinio objekto, ties kuriuo nupiešiamas rombas, bet ne pilnaviduris, o tuščiaviduris (žr. 8 pav.). [2, 7]



8 pav. Agregacijos ryšio pavyzdys

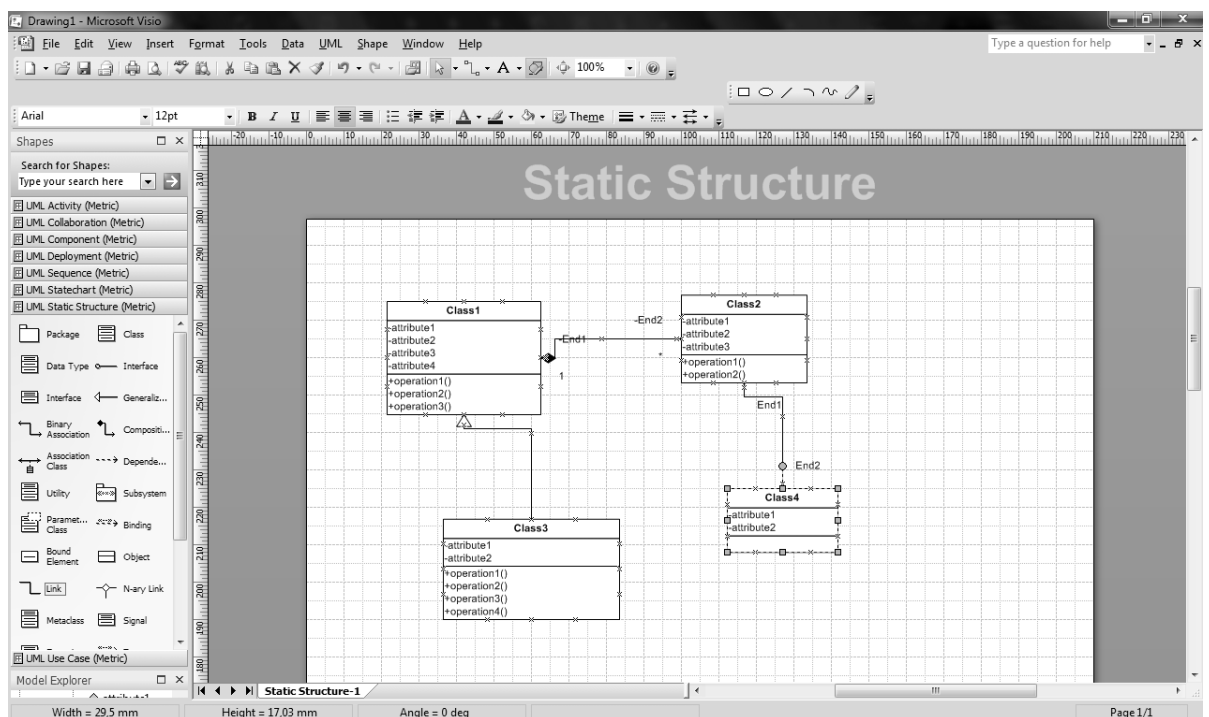
1.2 Klasių diagramos generavimo galimybės/programos

Klasių diagramos gali būti braižomos įvairiais būdais. Patartina visų pirma nusibraižyti klasių diagramą kuriamai programai ar sistemai ir tik tada pradėti pačios programos kūrimą, bet ne visada į tai yra atsižvelgiama dėl tam tikrų priežasčių. Pagrindinė priežastis kodėl yra pasirenkamas atvirkštinis variantas – programa projektuojama kūrimo metu. Aišku ne visada pavyksta nubraižyti tokią klasių diagramą kuri realiai atitiktų būsimą programą ir nereikėtų jokių pakeitimų.

Yra nemažai programų, kurių pagalba galima nesunkiai nubraižyti klasių diagramas prieš kuriant programos kodą. Tokios programos pagrįstos piešimo principu, kai pasirenkamas komponentas (šiuo atveju klasės blokas), nutempiamas į norimą vietą ekrane ir redaguojami atributai. Tokio tipo programos:

1.2.1 Microsoft Office Visio

Microsoft Office Visio – tai didelės apimties programa, skirta visokioms, tame tarpe ir UML, diagramoms braižyti. Ji turi kiekvienai diagramai priskirtą notaciją ir vadovaujasi „*Drag and drop*“ principu (minėtu anksčiau). Programa yra mokama.[3]

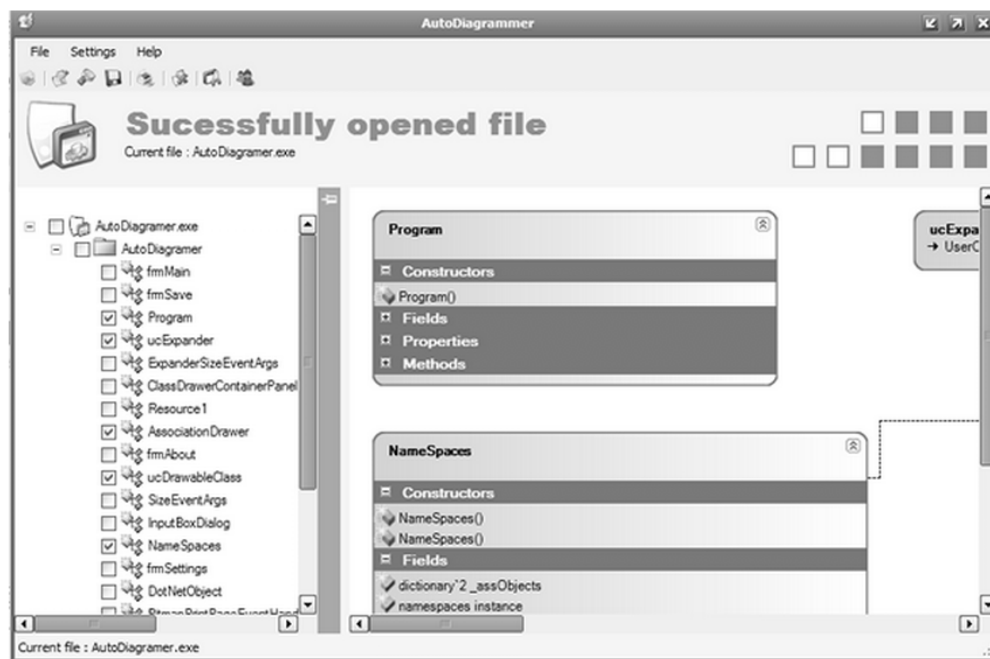


9 pav. Microsoft Office Visio 2007 programos vaizdas

Šia programa nėra sunku naudotis jeigu esi su ja susidūręs. Visi komponentai išdėstyti šoninėje programos dalyje, tereikia juos pasirinkti ir sujungti norimo tipo linijomis.

1. 2. 2 AutoDiagrammer

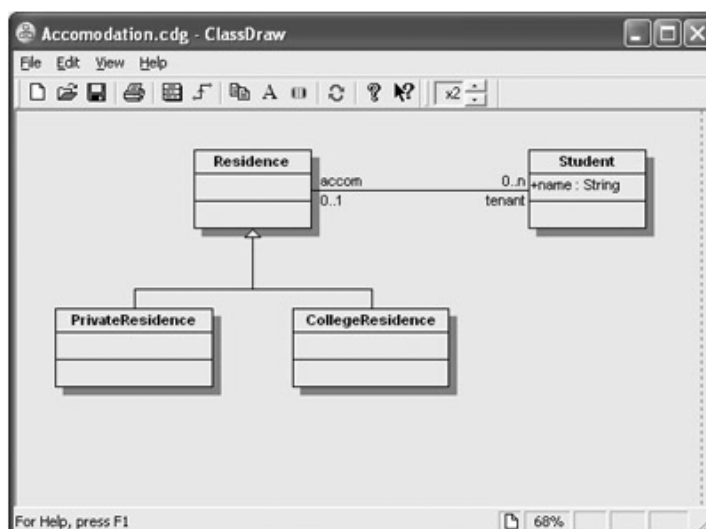
AutoDiagrammer – visiškai kitokio tipo nemokama programa. Ši programa skirta klasių diagramų braižymui iš parašyto programos kodo. Principas: paduodamas failas, kuriame yra programos kodas, šis failas nuskaitomas pagal programos kodo rašymo taisykles ir pagal nuskaitytą kodą sugeneruojama klasių diagrama. „AutoDiagrammer“ atpažįsta tik „.NET“, todėl netinka Java ar C++ kodo skaitymui. Visiškai skirtingai nei Microsoft Office Visio, ši programa skirta tik klasių diagramų braižymui ir užima palyginti labai nedaug vietos laikmenoje (~4MB). [6]



10 pav. „AutoDiagrammer“ programos vaizdas

1. 2. 3 ClassDraw

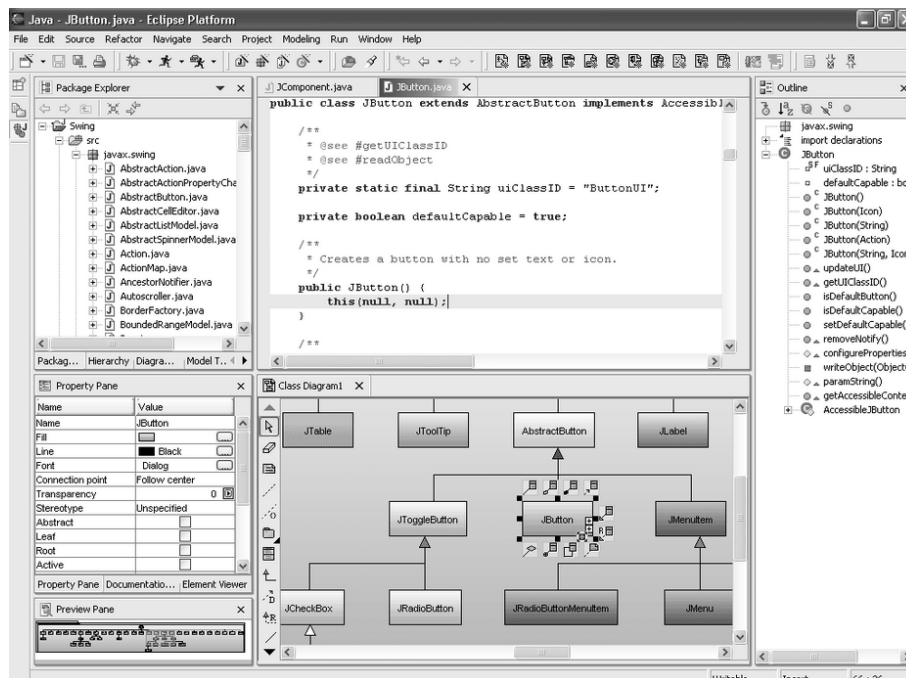
ClassDraw – taip pat nesudėtinga ir nedaug vietos užimanti programa, tik ji negeneruoja diagramų iš programos išeities kodo. Klasių diagramų braižymas pagrįstas „vedlių“ principu, tai kai pildomi laukai languose ir viską užpildžius gaunamas rezultatas. Deja ši programa piešia klasių elementus pati ir nesuteikia vartotojui galimybės keisti jų padėtį. Programos kaina 30 JAV dolerių. [1]



11 pav. „ClassDraw“ programos vaizdas

1. 2. 4 Eclipse

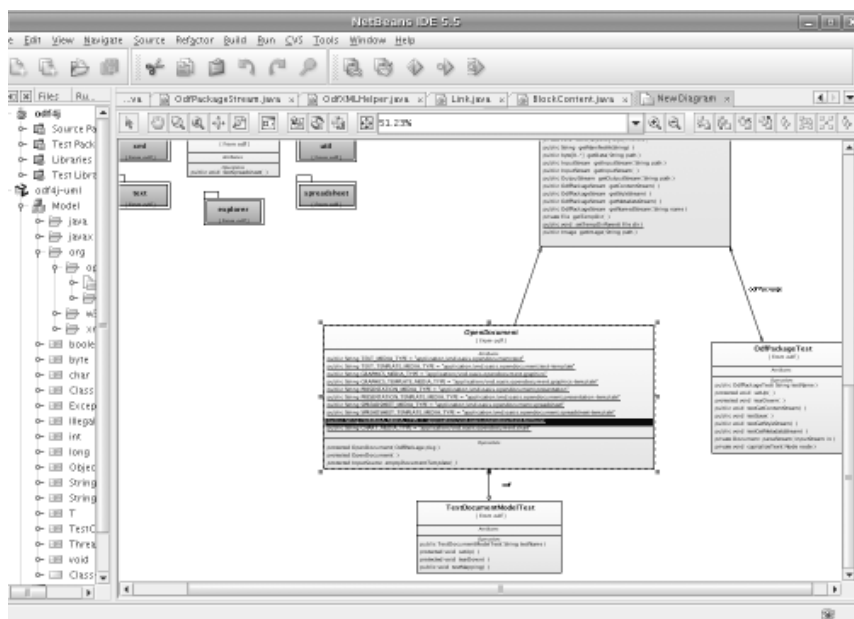
Eclipse – nemokama programa skirta programuotojams, tai programų kūrimo įrankis, tačiau turi galimybę iš parašyto programos kodo nubraižyti klasių diagramą. „Eclipse“ programa sukurtos diagramos praktiškai nebereikia redaguoti pačiam, nes ji iš kodo atskiria visa kas yra reikalinga diagramai atvaizduoti. Norint pasinaudoti šia galimybe, programos kodą reikia rašyti ta pačia programa, nes kaip atskiro modulio diagramoms piešti nepateikiama. [8]



12 pav. „Eclipse“ programos vaizdas

1. 2. 5 NetBeans

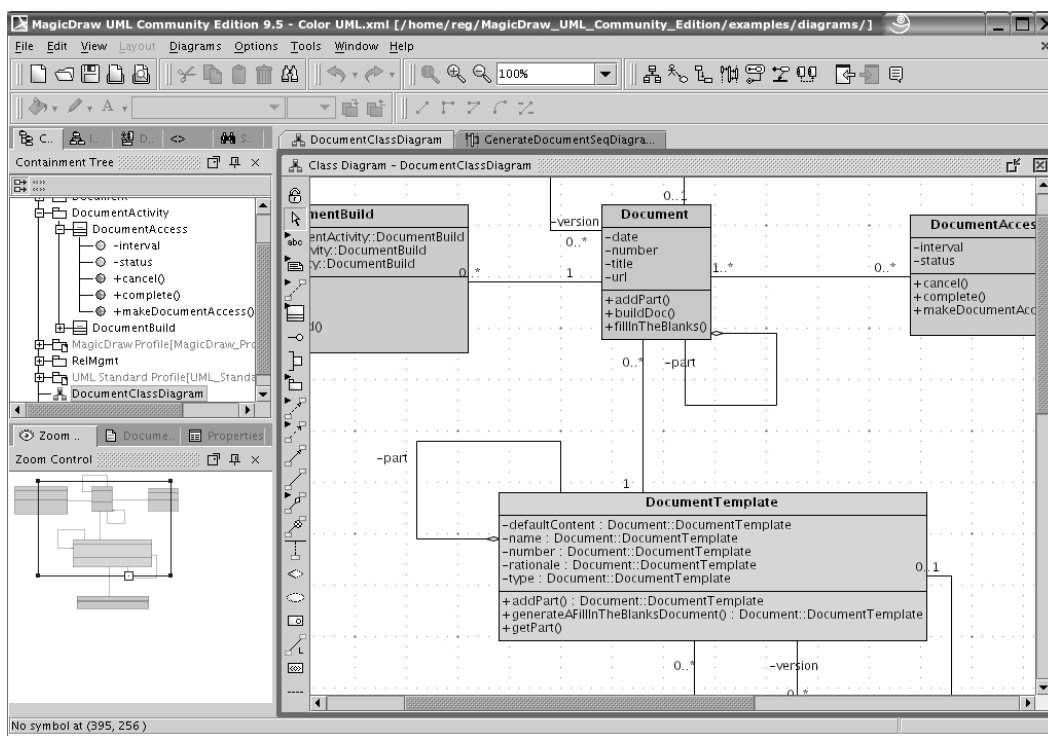
NetBeans – taip pat kaip ir „Eclipse“ skirta programų kūrimui ir turi panašų ir tokį pat tikslų modulį klasių diagramoms braižyti. Kaip ir „Eclipse“ šis modulis nėra panaudojamas atskirai nuo pačios programos, todėl reikia turėti visą „NetBeans“ platformą, kad būtų galima braižyti klasių diagramas. Ši programa yra nemokama. [5]



13 pav. „NetBeans“ programos vaizdas

1. 2. 6 MagicDraw

MagicDraw™ UML atitinka naujausius Java bei UML technologijų standartus, turi vieną iš patikimiausių išeities kodų inžinerijos mechanizmų Java, C#, C++ ir CORBA IDL programavimo kalboms bei gali vykdyti šių kalbų kodo atvirkštinę inžineriją, duomenų bazių schemų atvirkštinę inžineriją, kodo bei duomenų bazių schemų generavimą. MagicDraw™ UML yra vienas iš nedaugelio rinkoje esančių paketų, leidžiančių greičiau nubraižyti UML diagramas bei turintis UML diagramų semantinio teisingumo tikrinimo ir modelio validavimo mechanizmą. Ši programa yra mokama, o kaina priklauso nuo užsakomo paketo sudėties. [4]



14 pav. „MagicDraw“ programos vaizdas

1. 3 Programų palyginimas

1 lentelė. Programų palyginimas

Funkcija / Programa	MS Office Visio	Auto Diagrammer	ClassDraw	Eclipse	NetBeans	Magic Draw
Skirta tik klasių diagramų braižymui	Ne	Taip	Taip	Ne	Ne	Ne
Yra klasių diagramos generavimo iš šaltinio kodo funkcija	Ne	Taip	Ne	Taip	Taip	Taip
Skirta darbui su Java programavimo kalba	Ne	Ne	Ne	Taip	Taip	Taip
Ar nemokama?	Ne	Taip	Ne	Taip	Taip	Ne
Galimas sugeneruotos diagramos duomenų redagavimas	Ne	Taip	Ne	Taip	Taip	Taip
Galimas sugeneruotos diagramos išdėstymo redagavimas	Taip	Ne	Ne	Taip	Taip	Taip
Galimas sugeneruotos diagramos išvaizdos redagavimas	Ne	Ne	Ne	Ne	Ne	Ne

Vidinių klasių atpažinimo galimybė	Ne	Taip	Ne	Ne	Ne	Ne
Yra eksportavimo į paveikslėlio formatą galimybė	Taip	Taip	Taip	Taip	Taip	Taip
Generuojami ir ryšiai tarp klasių	Ne	Taip	Ne	Taip	Taip	Taip
Galima kurti klasių diagramą neturint programos kodo	Taip	Ne	Taip	Taip	Taip	Taip
Galima sąsaja lietuvių kalba	Ne	Ne	Ne	Ne	Ne	Ne
Daugiaplatformė	Ne	Ne	Ne	Taip	Taip	Ne
Generuojama iš konkretaus projekto, o ne visų parinktų failų	Ne	Ne	Ne	Taip	Taip	Ne
Užima mažiau nei 10MB vietos kompiuteryje	Ne	Taip	Taip	Ne	Ne	Ne

Visos išvardintos programos labiau ar mažiau tinka diagramų braižymui pagal vartotojo poreikius, bet vienu programų pagrindinis tikslas nėra diagramų braižymas, kitos per daug plataus diagramų tipo pasirinkimo, kas įtakoja programos sudėtingumą, dydį ir naudojamus kompiuterio resursus. Be to visos šios programos yra tik anglų kalba.

2. REIKALAVIMŲ SPECIFIKACIJA

2.1 Programinės įrangos paskirtis

2.1.1 Projekto kūrimo pagrindas

Klasių diagramų braižymo programos reikalingos palengvinti dokumentacijų ruošimo procesą ir jeigu jau turimas programos kodas tai iš jo automatinis klasių generavimas labai palengvina visą darbą. Kol kas rinkoje nėra analogiškų klasių diagramų braižymo programų, kurios būtų lietuvių kalba ir būtų skirtos tik tam tikslui, t.y. neturėtų kitų funkcijų, tik klasių diagramų braižymo.

2.1.2 Programinės įrangos tikslai, paskirtis

Sistema turi generuoti klasių diagramas iš pateiktų JAVA projekto failų kodo, jas išsaugoti paveikslėlio formatu (PNG) ir taip pat suteikti galimybę redaguoti išsaugotą klasių diagramą. Be to sistema turi būti itin lengvai valdoma, kad nereikėtų papildomai mokintis kaip naudotis.

2.2 Projekto dalyviai

2.2.1 Užsakovas

Asistentė Simona Ramanauskaitė, Šiaulių universitetas, Informacinių technologijų katedra, simram@it.su.lt.

2.2.2 Vykdytojas

I-6/2 grupės studentas Denas Pavlavičius, 869829216, denas.pavlavicius@gmail.com.

2.3 Vartotojai

2 lentelė. Vartotojo detalizavimas

Vartotojo kategorija	Vartotojas
Vartotojo sprendžiami uždaviniai:	Nubraižyti klasių diagramą pagal turimą programos išeities kodą, šią diagramą išsaugoti paveikslėlio formatu arba redaguojamu failu, kuri su ta pačia programa vėliau bus galima redaguoti.
Patirtis dalykinėje srityje:	Įprastas darbuotojas
Patirtis informacinėse technologijose:	Turintis pagrindines žinias
Papildomos vartotojo charakteristikos:	Reikalingos žinios apie klasių diagramas bei gebėjimas dirbti Windows aplinkoje
Apsimokymo poreikis:	Reikia
Amžiaus grupė:	19-80

2.4 Įpareigojantys apribojimai

2.4.1 Apribojimai sprendimui

- Nepriklausoma nuo sistemos;
- Skirta tik Java kalba parašyto programos kodo analizei ir atvaizdavimui klasių diagrama;
- Programa negeneruoja klasių diagramos iš klaidingai parašyto programos kodo ir neatidaro klaidingų „rkd“ failų.

2.4.2 Numatoma darbo vietos aplinka

Įprastinė darbo vieta turinti kompiuterį, atitinkanti ergonominius reikalavimus. Kompiuterio operacinėje sistemoje turi būti įdiegtas JAVA palaikymas (JVM).

2. 6. 2 Panaudos atvejų sąrašas

4 lentelė. Panaudojimo atvejo „Išrenkami JAVA kodų failai“ detalizavimas

1. PANAUDOJIMO ATVEJIS:	Išrenkami JAVA kodų failai
Vartotojas/Aktorius:	Vartotojas
Aprašas:	Apima procesą, kurio metu į sistemą importuojamas programos išeities kodas.
Prieš sąlyga:	Nėra informacijos pagal kurią reikėtų braižyti klasių diagramą
Sužadinimo sąlyga:	Atsirado Java kalba parašytas projekto kodas, kuriam norima nubraižyti/peržiūrėti klasių diagramą.
Po-sąlyga:	Sistema išsaugomi reikiami duomenys klasių diagramų braižymui.

5 lentelė. Panaudojimo atvejo „Paruošiama preliminarini diagrama“ detalizavimas

2. PANAUDOJIMO ATVEJIS:	Paruošiama preliminarini diagrama
Vartotojas/Aktorius:	Sistema
Aprašas:	Apima procesą, kurio metu nubraižoma preliminarini klasių diagrama
Prieš sąlyga:	Sistema turi informaciją apie būsimą klasių diagramą. Klasių diagrama nėra paruošta.
Sužadinimo sąlyga:	Vartotojo nurodytam Java projektui norima grafiškai peržiūrėti klasių diagramą.
Po-sąlyga:	Sistema paruošia preliminarini klasių diagramą

6 lentelė. Panaudojimo atvejo „Redaguoti“ detalizavimas

3. PANAUDOJIMO ATVEJIS:	Redaguoti
Vartotojas/Aktorius:	Vartotojas
Aprašas:	Apima procesą, kurio metu vartotojas gali redaguoti klasių diagramą
Prieš sąlyga:	Paruošta preliminarini klasių diagrama.
Sužadinimo sąlyga:	Vartotojas paspaudžia nuorodą „Redaguoti“
Po-sąlyga:	Preliminarini klasių diagrama gali būti redaguojama

7 lentelė. Panaudojimo atvejo „Klasių išdėstymo redagavimas“ detalizavimas

4. PANAUDOJIMO ATVEJIS:	Klasių išdėstymo redagavimas
Vartotojas/Aktorius:	Vartotojas
Aprašas:	Apima procesą, kurio metu redaguojamas klasių išdėstymas diagramoje.
Prieš sąlyga:	Paruošta preliminarini klasių diagrama.
Sužadinimo sąlyga:	Vartotojas pasirenka klasę ir ją pradeda „tempti“
Po-sąlyga:	Redaguojamas klasių išdėstymas

8 lentelė. Panaudojimo atvejo „Ryšių tarp klasių redagavimas“ detalizavimas

5. PANAUDOJIMO ATVEJIS:	Ryšių tarp klasių redagavimas
Vartotojas/Aktorius:	Vartotojas
Aprašas:	Apima procesą, kurio metu redaguojami ryšiai tarp klasių
Prieš sąlyga:	Paruošta preliminarini klasių diagrama.
Sužadinimo sąlyga:	Vartotojas spaudžia dešinį pelės klavišą ant klasės ir išrenka ryšių redagavimo punktą iš meniu.
Po-sąlyga:	Redaguojami ryšiai tarp klasių.

9 lentelė. Panaudojimo atvejo „Bendro vaizdo redagavimas“ detalizavimas

6. PANAUDOJIMO ATVEJIS:	Bendro vaizdo redagavimas
Vartotojas/Aktorius:	Vartotojas
Aprašas:	Apima procesą, kurio metu redaguojamas bendras klasės vaizdas
Prieš sąlyga:	Paruošta klasių diagrama.
Sužadinimo sąlyga:	Vartotojas pasirenka klasės šrifto, spalvos ir dydžio redagavimo punktą
Po-sąlyga:	Pritaikytas norimas klasės atvaizdavimas

10 lentelė. Panaudojimo atvejo „Klasės parametrų redagavimas“ detalizavimas

7. PANAUDOJIMO ATVEJIS:	Klasės parametrų redagavimas
Vartotojas/Aktorius:	Vartotojas
Aprašas:	Apima procesą, kurio metu redaguojami klasės parametrai
Prieš sąlyga:	Paruošta klasių diagrama.
Sužadinimo sąlyga:	Vartotojas spaudžia dešinį pelės klavišą ant klasės ir pasirenka parametrų redagavimo punktą.
Po-sąlyga:	Pakeičiami klasės parametrai

11 lentelė. Panaudojimo atvejo „Išsaugojimas“ detalizavimas

8. PANAUDOJIMO ATVEJIS:	Išsaugojimas
Vartotojas/Aktorius:	Vartotojas
Aprašas:	Apima procesą, kurio metu išsaugoma klasių diagrama
Prieš sąlyga:	Paruošta klasių diagrama.
Sužadinimo sąlyga:	Vartotojas pasirenka išsaugojimo nuorodą
Po-sąlyga:	Klasių diagrama išsaugota atmintyje

12 lentelė. Panaudojimo atvejo „Išsaugojimas redaguojamu failu“ detalizavimas

9. PANAUDOJIMO ATVEJIS:	Išsaugojimas redaguojamu failu
Vartotojas/Aktorius:	Vartotojas
Aprašas:	Apima procesą, kurio metu klasių diagrama išsaugoma atmintyje failu, kurį bus galima redaguoti su ta pačia sistema.
Prieš sąlyga:	Paruošta klasių diagrama.
Sužadinimo sąlyga:	Vartotojas pasirenka redaguojamą tipą
Po-sąlyga:	Klasių diagrama išsaugota tolesniam redagavimui

13 lentelė. Panaudojimo atvejo „Išsaugojimas paveikslėlio formatu“ detalizavimas

10. PANAUDOJIMO ATVEJIS:	Išsaugojimas paveikslėlio formatu
Vartotojas/Aktorius:	Vartotojas
Aprašas:	Apima procesą, kurio metu klasių diagrama išsaugojama paveikslėlio formatu
Prieš sąlyga:	Paruošta klasių diagrama.
Sužadinimo sąlyga:	Vartotojas pasirenka paveikslėlio formato tipą
Po-sąlyga:	Klasių diagrama išsaugota paveikslėlio formatu

14 lentelė. Panaudojimo atvejo „Diagramos atvaizdavimas“ detalizavimas

11. PANAUDOJIMO ATVEJIS:	Diagramos atvaizdavimas
Vartotojas/Aktorius:	Vartotojas
Aprašas:	Apima procesą, kurio metu klasių diagrama atvaizduojama ekrane
Prieš sąlyga:	Atidarytas RKD failas arba sugeneruota klasių diagrama.
Sužadinimo sąlyga:	Vartotojas atidaro RKD failą arba importuoja JAVA failus
Po-sąlyga:	Klasių diagrama atvaizduota ekrane

15 lentelė. Panaudos atvejo „Diagramos atidarymas iš RKD failo“ detalizavimas

12. PANAUDOJIMO ATVEJIS:	Diagramos atidarymas iš RKD failo
Vartotojas/Aktorius:	Vartotojas
Aprašas:	Apima procesą, kurio metu klasių diagramos duomenys nuskaitomi iš failo
Prieš sąlyga:	Norima atidaryti išsaugotą klasių diagramą.
Sužadinimo sąlyga:	Vartotojas pasirenka RKD formato failą
Po-sąlyga:	Klasių diagrama atvaizduota ekrane

2.7 Funkciniai reikalavimai

16 lentelė. Funkcinio reikalavimo „JAVA failų parinkimas“ detalizavimas

Reikalavimas #:	1	Reikalavimo tipas:	2.7	Ivykis/panaudojimo atvejis #:	1
Aprašymas:	Sistema turi leisti vartotojui pasirinkti direktoriją su JAVA kodų failais				
Pagrindimas:	Norint, kad sistema veiktų, jai reikia perduoti JAVA išeities kodų failus, nes be jų sistema negalės sugeneruoti klasių diagramų.				
Šaltinis:	Projekto vadovas				
Tikimo kriterijus:	Vartotojas nurodo sistemai katalogą, o sistema iš to katalogo išrenka visus failus su plėtiniu „java“.				
Užsakovo tenkinimas:	3	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		

17 lentelė. Funkcinio reikalavimo „Diagramos generavimas“ detalizavimas

Reikalavimas #:	2	Reikalavimo tipas:	2.7	Ivykis/panaudojimo atvejis #:	2
Aprašymas:	Sistema turi sugeneruoti preliminarą klasių diagramą iš nurodyto išeities kodo.				
Pagrindimas:	Vartotojas turi pamatyti bendrą vaizdą prieš atliekant kitus redagavimo veiksmus.				
Šaltinis:	Projekto vadovas				
Tikimo kriterijus:	Ekrane parodoma preliminarinė klasių diagrama, tai yra klasių išdėstymas, ryšiai tarp jų ir informacija apie klases, jei nurodytas programos išeities kodas užrašytas JAVA kalba.				
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	5		
Priklausomybės:	1	Konfliktai:	Nėra		

18 lentelė. Funkcinio reikalavimo „Klasių diagramos redagavimas“ detalizavimas

Reikalavimas #:	3	Reikalavimo tipas:	2.7	Ivykis/panaudojimo atvejis #:	4
Aprašymas:	Sistema turi leisti vartotojui redaguoti klasių grafinį išdėstymą				
Pagrindimas:	Vartotojo gali netenkinti klasių išdėstymas, arba jis gali būti painus				
Šaltinis:	Projekto vadovas				
Tikimo kriterijus:	Vartotojas gali pernešti klasės bloką iš vienos ekrano vietos į kitą.				
Užsakovo tenkinimas:	2	Užsakovo netenkinimas:	5		
Priklausomybės:	2	Konfliktai:	Nėra		

19 lentelė. Funkcinio reikalavimo „Ryšių redagavimas“ detalizavimas

Reikalavimas #:	4	Reikalavimo tipas:	2.7	Ivykis/panaudojimo atvejis #:	5
Aprašymas:	Sistema turi leisti vartotojui redaguoti ryšius tarp klasių				
Pagrindimas:	Sistema gali būti nepajėgi nurodyti automatiškai visų esamų ryšių, arba vartotojas nori juos pakeisti				
Šaltinis:	Projekto vadovas				
Tikimo kriterijus:	Vartotojui leidžiama pasirinkti klasės ryšius ir juos redaguoti, t.y. keisti jų tipą, kardinalumą.				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	5		
Priklausomybės:	2	Konfliktai:	Nėra		

20 lentelė. Funkcinio reikalavimo „Klasės parametrų redagavimas“ detalizavimas

Reikalavimas #:	5	Reikalavimo tipas:	2.7	Ivykis/panaudojimo atvejis #:	5
Aprašymas:	Sistema turi leisti vartotojui redaguoti klasės parametrus				
Pagrindimas:	Esant vartotojo poreikiui turi būti galimybė panaikinti klasės atributus arba pridėti kitus.				
Šaltinis:	Projekto vadovas				
Tikimo kriterijus:	Vartotojui leidžiama pasirinkti klasę ir redaguoti jos parametrus.				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	5		
Priklausomybės:	2	Konfliktai:	Nėra		

21 lentelė. Funkcinio reikalavimo „Klasių diagramos saugojimas paveikslėlio formatu“ detalizavimas

Reikalavimas #:	6	Reikalavimo tipas:	2.7	Ivykis/panaudojimo atvejis #:	8
Aprašymas:	Sistema turi leisti vartotojui išsaugoti klasių diagramą paveikslėlio formatu				
Pagrindimas:	Paveikslėlio formate esanti klasių diagrama yra labai patogi tolimesniems jos naudojimams ar publikavimui.				
Šaltinis:	Projekto vadovas				
Tikimo kriterijus:	Vartotojui leidžiama išsaugoti sugeneruotą klasių diagramą paveikslėlio (*.png) formatu.				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	5		
Priklausomybės:	2	Konfliktai:	Nėra		

22 lentelė. Funkcinio reikalavimo „Klasių diagramos saugojimas redaguojamu formatu“ detalizavimas

Reikalavimas #:	7	Reikalavimo tipas:	2.7	Ivykis/panaudojimo atvejis #:	7
Aprašymas:	Sistema turi leisti vartotojui išsaugoti klasių diagramą redaguojamu formatu				
Pagrindimas:	Jeigu ateityje įvyktų kokie nors nežymūs pasikeitimai, nebereiktų kurti klasių diagramos iš naujo				
Šaltinis:	Projekto vadovas				
Tikimo kriterijus:	Vartotojui leidžiama išsaugoti redaguojamo failo formatu, t.y. failu su plėtinium „rkd“.				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	5		
Priklausomybės:	3, 4	Konfliktai:	Nėra		

23 lentelė. Funkcinio reikalavimo „Naujos klasių diagramos kūrimas“ detalizavimas

Reikalavimas #:	8	Reikalavimo tipas:	2.7	Ivykis/panaudojimo atvejis #:	1
Aprašymas:	Sistema turi leisti vartotojui kurti naują klasių diagramą				
Pagrindimas:	Jeigu vartotojas yra jau pradėjęs kurti klasių diagramą, turi būti galimybė pradėti kurti naują.				
Šaltinis:	Projekto vadovas				
Tikimo kriterijus:	Vartotojui leidžiama pradėti naują diagramą				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		

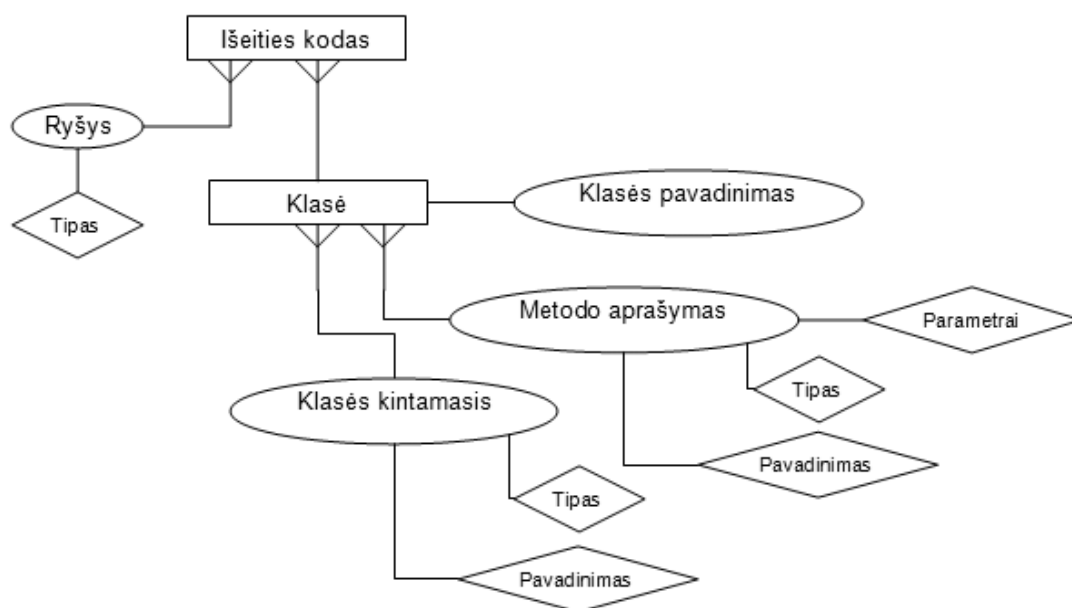
24 lentelė. Funkcinio reikalavimo „Ryšio nustatymas“ detalizavimas

Reikalavimas #:	9	Reikalavimo tipas:	2.7	Ivykis/panaudojimo atvejis #:	5
Aprašymas:	Sistema turi nustatyti ryšį tarp dviejų klasių (paveldėjimas, kompozicija, agregacija ir asociacija)				
Pagrindimas:	Reikia pirma atskirti koks tai ryšys norint jį pavaizduoti.				
Šaltinis:	Projekto vadovas				
Tikimo kriterijus:	Sistema automatiškai peržiūri programos išeities kodą ir nustato kokie ryšiai yra tarp klasių.				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	5		
Priklausomybės:	1	Konfliktai:	Nėra		

25 lentelė. Funkcinio reikalavimo „Ryšų kardinalumo nustatymas“ detalizavimas

Reikalavimas #:	10	Reikalavimo tipas:	2.7	Ivykis/panaudojimo atvejis #:	5
Aprašymas:	Sistema turi nustatyti ryšių kardinalumą, t.y. viršutinę ir apatinę ryšio naudojimo ribas.				
Pagrindimas:	Norint atvaizduoti kardinalumą, reikia jį nustatyti				
Šaltinis:	Projekto vadovas				
Tikimo kriterijus:	Sistema automatiškai peržiūri programos išeities kodą ir kur įmanoma nustato kardinalumą.				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	5		
Priklausomybės:	1	Konfliktai:	Nėra		

2. 7. 1 Reikalavimai duomenims



17 pav. Sistemos duomenų loginė struktūra

2. 8 Nefunkciniai reikalavimai

2. 8. 1 Reikalavimai sistemos išvaizdai

26 lentelė. Nefunkcinio reikalavimo „Vartotojo sąsaja“ detalizavimas

Reikalavimas #:	11	Reikalavimo tipas:	2.8.1	Ivykis/panaudojimo atvejis #:	-
Aprašymas:	Sistemos sąsaja turi būti lengvai skaitoma.				
Pagrindimas:	Sistemos sąsaja turi būti lengvai matoma, žiūrima, skaitoma bei suprantama, tam kad programa būtų patogiu naudotis.				
Šaltinis:	Projekto vadovas				
Tikimo kriterijus:	Sistemos sąsaja lengvai suprantama ir nesunkiai valdoma				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		

2. 8. 2 Reikalavimai panaudojamumui

27 lentelė. Nefunkcinio reikalavimo „Galimos sistemos klaidos“ detalizavimas

Reikalavimas #:	12	Reikalavimo tipas:	2.8.2	Ivykis/panaudojimo atvejis #:	
Aprašymas:	Sistema turi būti tolerantiška galimoms klaidoms				
Pagrindimas:	Vartotojas gali padaryti klaidų dirbant su programa, todėl sistema turi būti kiek įmanoma tolerantiškesnė galimoms klaidoms.				
Šaltinis:	Projekto vadovas				
Tikimo kriterijus:	Sistema tolerantiška klaidoms.				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		

28 lentelė. Nefunkcinio reikalavimo „Sistemos naudojimo nesudėtingumas“ detalizavimas

Reikalavimas #:	13	Reikalavimo tipas:	2.8.2	Ivykis/panaudojimo atvejis #:	1-8
Aprašymas:	Sistema turi būti nesudėtinga išmokti naudotis				
Pagrindimas:	Neturėtų būti poreikis vartotojams baigti spec. kursus apie naudojimąsi programa.				
Šaltinis:	Projekto vadovas				
Tikimo kriterijus:	Vartotojas išmoksta dirbti su sistema per vieną dieną				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	4		
Priklausomybės:	Nėra	Konfliktai:	Nėra		

29 lentelė. Nefunkcinio reikalavimo „Sistemos veikimas OS“ detalizavimas

Reikalavimas #:	14	Reikalavimo tipas:	2.8.2	Ivykis/panaudojimo atvejis #:	
Aprašymas:	Sistema turi veikti Microsoft Windows XP operacinėje sistemoje.				
Pagrindimas:	Užsakovas ruošiasi naudoti programą Microsoft Windows XP operacinėje sistemoje.				
Šaltinis:	Projekto vadovas				
Tikimo kriterijus:	Sistema veikia Microsoft Windows XP operacinėje sistemoje				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		

2. 8. 3 Reikalavimai vykdymo charakteristikoms

30 lentelė. Nefunkcinio reikalavimo „Atvaizdavimo greitis“ detalizavimas

Reikalavimas #:	15	Reikalavimo tipas:	2.8.3	Ivykis/panaudojimo atvejis #:	
Aprašymas:	Programa turi atvaizduoti klasių diagramą ne ilgiau kaip per 500 ms.				
Pagrindimas:	Ilgesnis laiko tarpas gali trikdyti vartotojo darbą.				
Šaltinis:	Projekto vadovas				
Tikimo kriterijus:	Programa atvaizduoja klasių diagramą ne daugiau kaip per 500 ms.				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		

31 lentelė. Nefunkcinio reikalavimo „Naudojami resursai“ detalizavimas

Reikalavimas #:	16	Reikalavimo tipas:	2.8.3	Ivykis/panaudojimo atvejis #:	
Aprašymas:	Programa negali naudoti daugiau nei 1% sistemos resursų.				
Pagrindimas:	Sunaudojant daugiau resursų galima sukelti nepatogumų vartotojui.				
Šaltinis:	Projekto vadovas				
Tikimo kriterijus:	Programa nenaudoja daugiau nei 1% sistemos resursų.				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		

2. 8. 4 Reikalavimai saugumui

32 lentelė. Nefunkcinio reikalavimo „Vientisumas“ detalizavimas

Reikalavimas #:	17	Reikalavimo tipas:	2.8.4	Ivykis/panaudojimo atvejis #:	
Aprašymas:	Sistemos vientisumas.				
Pagrindimas:	Sistemos duomenys vienareikšmiškai atitinka šaltinio perduotus (iš jo gautus) duomenis, kartu užtikrinant jų panaudojimo teisėtumą.				
Šaltinis:	Projekto vadovas				
Tikimo kriterijus:	Perduoti sistemai duomenys nėra iškraipomi				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	5		
Priklausomybės:	Nėra	Konfliktai:	Nėra		

2. 9 Atviri klausimai

- Kas bus jeigu pasikeis klasių diagramos notacija?
- Kokio maksimalaus sudėtingumo klasių diagramą gali nubraižyti programa?

2. 10 Egzistuojantys sprendimai

2. 10. 1 Galimi panaudoti komponentai

Programoje gali būti panaudota nemokama „NetBeans“ vidinė biblioteka – „beansbinding-1.2.1“, kuri skirta programos objektų ir visualinių komponentų tarpusavio susiejimui.

2.11 Pritaikymas

2.11.1 Reikalavimai esamų duomenų perkėlimui

Klasių diagrama bus saugoma dviem būdais:

- Paveikslėlio formatu, kurio plėtinys bus „png“. Šį paveikslėlį bus galima peržiūrėti bet kur, kur tik įmanoma paveikslėlių peržiūra.
- Redaguojamu formatu, kurio plėtinys bus „rkd“. Šį failą galima perkelti ir peržiūrėti tik tokią pat programą turinčioje sistemoje.

2.12 Galimos sistemos kūrimo rizikos

33 lentelė. Sistemos kūrimo rizikos

Rizikos faktorius	Tikimybinis įvertinimas
Reikalavimų specifikacijos pasikeitimai realizavimo fazėje	10
Nauja klasių diagramos notacija	3
Neatstatomas klasių išdėstymas atidarius redaguojamą klasių diagramą	5
Netvarkingas automatinis klasių išdėliojimas ekrane	5
Netiksli klasių diagrama dėl pernelyg sudėtingo programos išeities kodo	5

2.13 Vartotojo dokumentacija ir apmokymas

Vartotojui pateikiamas vartotojo bei diegimo vadovai.

2.14 Perspektyviniai reikalavimai

- Generuoti klasių diagramas ir iš kitų programavimo kalbų išeities kodą.
- Iš pakeistos klasių diagramos generuoti programos kodą.

3. PROGRAMINĖS ĮRANGOS ARCHITEKTŪROS SPECIFIKACIJA

3.1 Architektūros specifikacijos paskirtis

Architektūros specifikacija siekiama supažindinti su būsimu produkto veikimo principu. Pateikti programos veikimo etapus bei galimybes. Parengti išsamų architektūrinį aprašymą vartotojui ir veikimo specifikaciją programuotojui. Informuoti apie architektūros tikslus ir apribojimus.

3.2 Architektūros pateikimas

Klasių diagramų braižymo programos architektūra yra pateikiama diagramomis bei jų trumpais apibūdinimais. Žemiau pateiktos veiklos, sekų, būsenų, bendradarbiavimo diagramos. Taip pat pateikta klasių diagrama. Diagramomis siekiama iliustruoti kaip duomenys sąveikauja tarpusavyje, kas ir kaip su jais dirba. Kaip programa gauna informaciją apie būsimą klasių diagramą ir kokių principu turi ją nubraižyti. Kada ir kaip sąveikauja programos objektai tarpusavyje ir kokius duomenis perduoda vieni kitiems.

3.3 Architektūros tikslai ir apribojimai

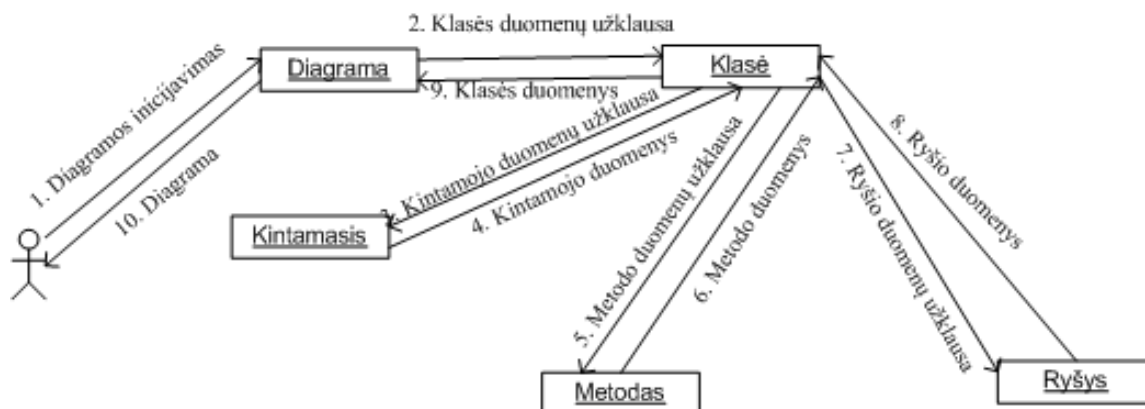
Kuriant programą neturėtų būti naudojamos papildomos bibliotekos, kurios sumažintų programos panaudojimą kitose sistemose. Bei neturėtų būti naudojamos išorinės duomenų bazės, taip programą padarant ne tokią priklausomą nuo kitų sistemų.

Vienintelis reikalavimas sistemoje turi būti įdiegta Java Virtuali Mašina (JVM 6 <http://java.com/en/download/manual.jsp>).

3.4 Sistemos dinaminis vaizdas

3.4.1 Programos objektų tarpusavio bendradarbiavimas

Objektų tarpusavio bendradarbiavimas parodo kaip objektai bendrauja ir kokius pranešimus vieni kitiems siunčia. Vartotojas tiesiogiai bendrauja su pagrindiniu programos objektu „diagrama“, kuris savo ruožtu bendrauja su klasės objektu. Klasės objektas pateikia užklausas kintamojo, metodo ir ryšio objektams apie jų duomenis, surinkti duomenys perduodami diagramos objektui, kuri vartotojui juos pateikia vaizdine išraiška, t.y. klasių diagrama. (žr. 18 pav.)

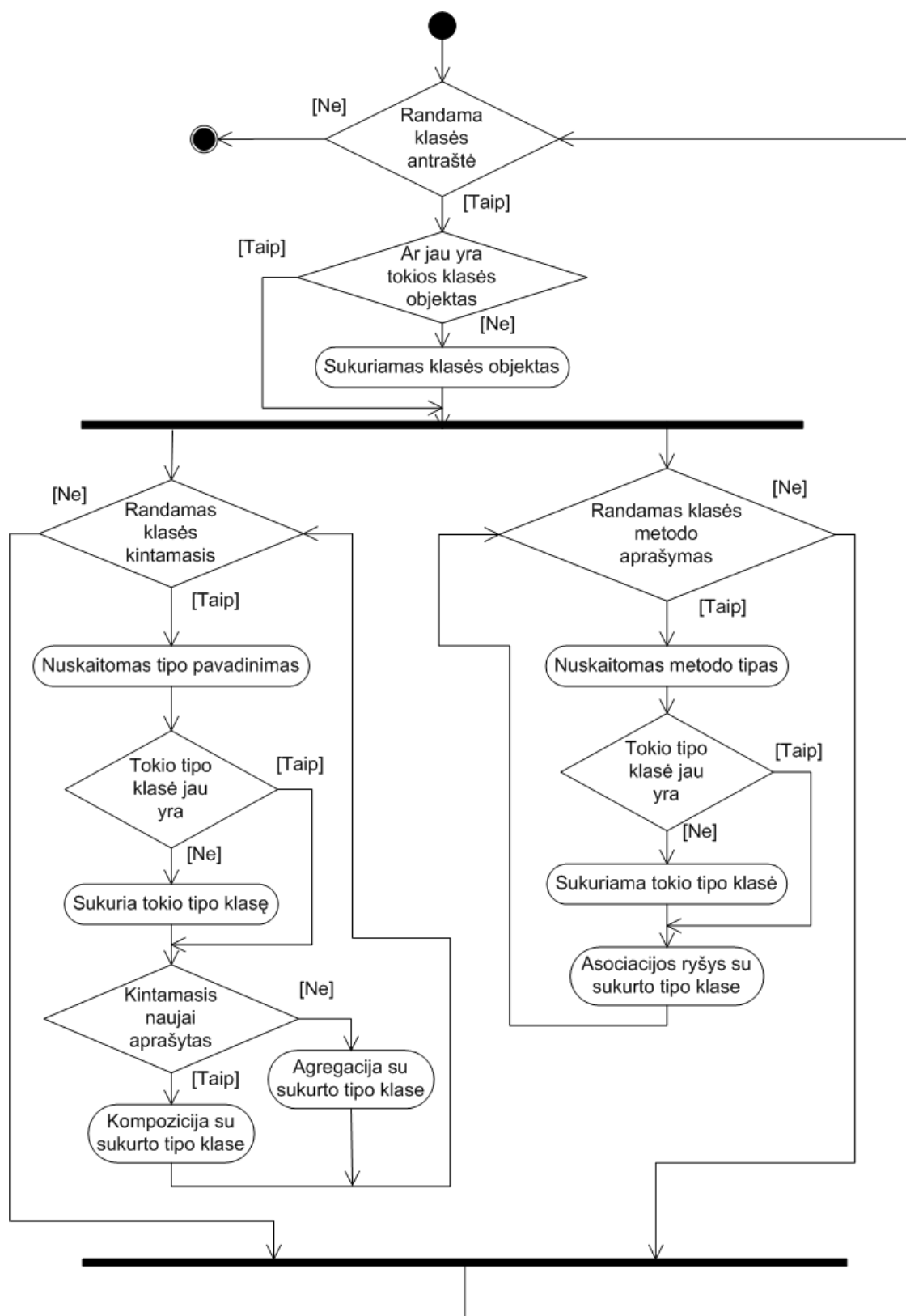


18 pav. Programos objektų tarpusavio bendradarbiavimo diagrama

3.4.2 Duomenų surinkimas

Duomenų surinkimas vyksta keliais etapais, kol sistemoje užregistruojama informacija apie klases. Pirmas etapas – klasės aptikimas. Kai aptinkama klasė, tada pereinama prie klasės kintamųjų ir metodų paieškos. Radus kintamąjį jis yra užregistruojamas sistemoje. Iš kintamojo tipo galima

nuspręsti apie klasės ryšius su kitomis klasėmis, todėl kintamojo tipas yra registruojamas kaip atskira klasė, arba ieškoma jau užregistruotos. Metodų paieška vykdoma panašiai tik tikrinamas metodo grąžinamas tipas ir atliekami tokie pat veiksmai kaip ir su kintamojo tipu. Kai visi kintamieji ir metodai priklausantys tai klasei surinkti, tada ieškoma sekančios klasės ir taip tol kol visos klasės bus užregistruotos (žr. 19 pav.).

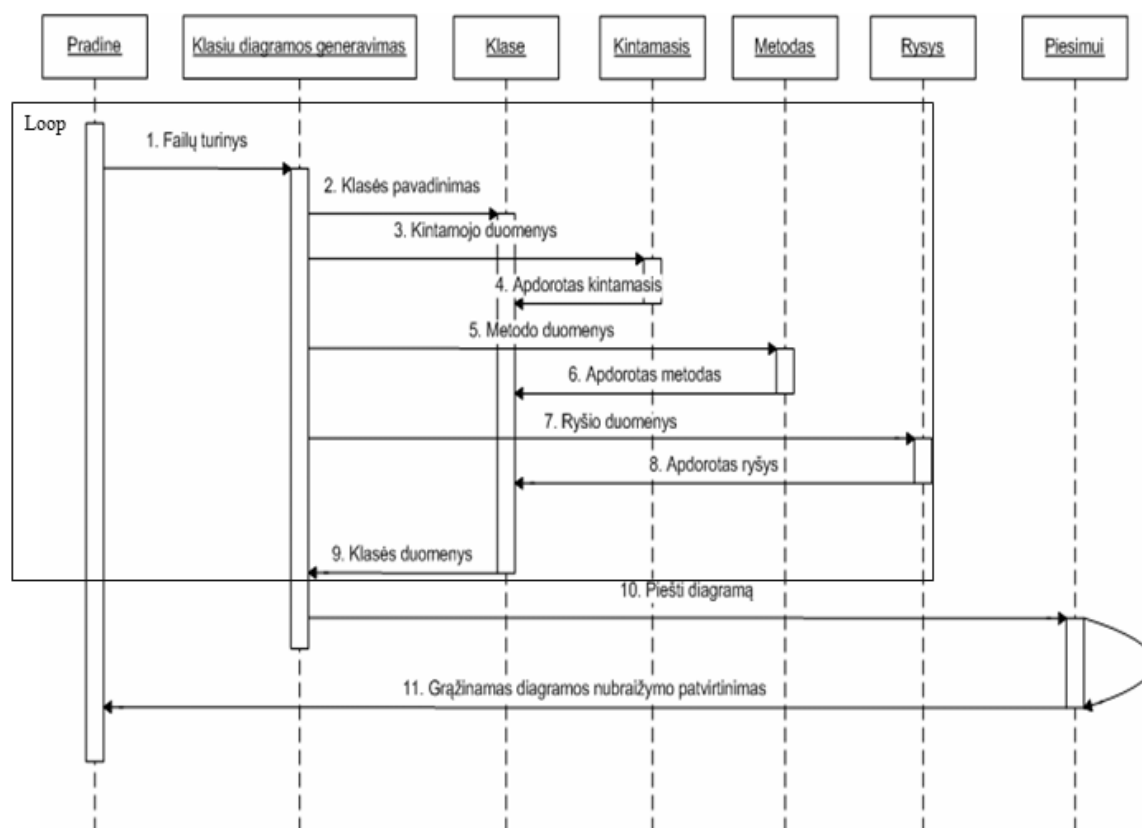


19 pav. Duomenų surinkimo veiklos diagrama

3. 4. 3 Klasų diagramos generavimas

Klasų diagramos generavimas vyksta nuosekliai. Visų pirma programai perduodami „java“ failai su programos šaltinio kodu. Tada programa eina per gautą kodą ir ieško klasės aprašymo, radus klasės aprašymą sukuriamas naujas klasės objektas ir jam perduodamas rastos klasės pavadinimas. Inicijavus klasę programa pradeda darbą jos viduje. Rastas kintamojo aprašymas paverčiamas kintamojo objektu ir į jį surašoma visa informacija. Toliau seka metodo aprašymas, kurį radus sukuriamas metodo objektas. Pagal rastus kintamųjų bei metodų tipus nustatomi ryšiai ir sukuriama ryšio objektai į kuriuos taip pat patalpinama informacija apie ryšį. Baigus analizuoti klasę, programa uždaro sukurtą klasės objektą ir pereina prie sekančios klasės, ir taip tol kol visos klasės būna aprašytos.

Kai jau visa informacija surinkta klasų diagramos generavimo objektas perduoda klasės objektą piešimo objektui, kuris savo ruožtu pagal gautą informaciją nupiešia klasų diagramą ir grąžina pagrindiniam klasės objektui patvirtinimą, kad klasų diagrama nubraižyta (žr. 20 pav.).

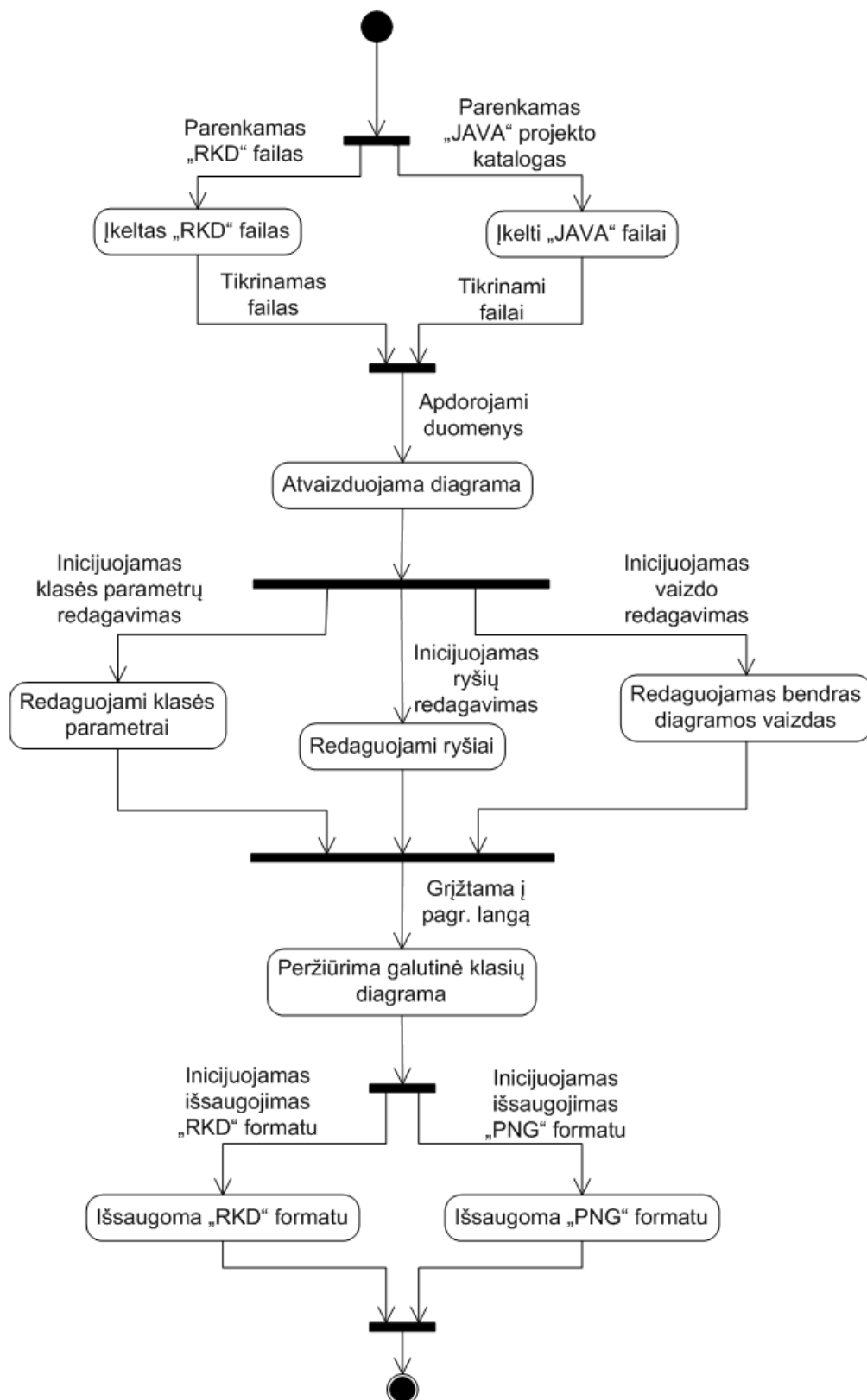


20 pav. Klasų diagramos generavimo sekų diagrama

3. 4. 4 Programos būsenos

Programos būsena kinta nuo pateikto programos išeities kodo iki išsaugotos klasės diagramos. Įvykiai, kurie turi įtakos būsenoms: atidarius programą, norint inicijuoti programos darbą reikia jai pateikti savo projekto „java“ failus arba atverti anksčiau išsaugotą „rkd“ failą. Jei viskas padaryta teisingai, programa ekrane atvaizduos klasų diagramą. Tada vartotojas gali inicijuoti pasirinktinai ryšių, klasės parametrų ar bendro vaizdo redagavimą. Baigus redagavimą grįžtama į pagrindinį programos langą, kuriame peržiūrima paredaguota klasų diagrama. Jei vartotoją tenkina matomas vaizdas, tada inicijuojamas saugojimas pasirinktinai „png“ (paveikslėlio) ar „rkd“ (sistemos failo)

formatu. Kai visi veiksmai atlikti vartotojas turi išsaugotą klasių diagramą iš savo pateiktų „java“ kodų (žr. 21 pav.).



21 pav. Klasių diagramų braižymo programos būsėnų diagrama

3.5 Sistemos statinis vaizdas

Sistemoje naudojami šie objektai: klase, metodas, tipas, kintamasis, rysys, keturkampis, pradine, klasiuDiagramosgeneravimas, piesimui. Visi šie objektai vienaip ar kitaip tarpusavyje yra susiję. Objektas „Klase“ yra kompoziciniu būdu susijęs su objektais metodas, kintamasis ir rysys, taip

```

classDiagram
    class Metodas {
        +pavadinimas: String
        +grazinamasTipas: Tipas
        +parametrai: ArrayList<Kintamasis>
        +matomumas: int
        +statinis: boolean
        +metodoFormatas(): String
    }
    class Tipas {
        +klase: Klase
        +masyvas: boolean
    }
    class Kintamasis {
        +matomumas: int
        +statinis: boolean
        +pavadinimas: String
        +tipas: Tipas
        +kiekis: int
        +kompozicinis: boolean
        +kintamojoFormatas(): String
    }
    class Keturkampis {
        +maxX: int
        +x_nuo: int
        +y_nuo: int
        +aukstis: int
        +aukstis2: int
        +plotis: int
        +keistiPadeti(int xp, int yp): void
        +ieina(int x, int y): boolean
    }
    class Klase {
        +pavadinimas: String
        +kintamieji: ArrayList<Kintamasis>
        +metodai: ArrayList<Metodas>
        +ryusiai: ArrayList<Rysys>
        +projektoKlase: boolean
        +pridetiKintamaji(Kintamasis k): void
        +pridetiMetoda(Metodas m): void
        +pridetiRysyi(Rysys r): void
        +perskaiciuotiTeritorija(): void
        +skaiciuotiMatmenis(int dydis): void
    }
    class Rysys {
        +tipas: int
        +kardinalumas: int
        +klase: Klase
        +rysysPer: Kintamasis
    }
    class Pradine {
        +bnd: KlasiuDiagramosGeneravimas
        +krastas: boolean
        +listas: ArrayList<Klase>
        +piesimui1: atvaizdavimas.Piesimui
        +loadFile(Frame f, String title, String defDir, String fileType): String
        +saveFile(Frame f, String title, String defDir, String fileType): String
    }
    class KlasiuDiagramosGeneravimas {
        +projektoKlasiuAdresas: ArrayList<String>
        +klasiuSarasas: ArrayList<Klase>
        +tipuSarasas: ArrayList<Tipas>
        +dabartine: Klase
        +KlasiuDiagramosGeneravimas(File projektoAdresas): void
        +gautiKlasiuSarasa(File dir): void
        +analizuotiVisaProjekta(): void
        +arKabutese(String elute, int vieta): boolean
        +analizuotiFailoKoda(String kodas): void
        +analizuotiAntraste(String kodas): void
        +gautiKlase(String pav): Klase
        +analizuotiKintamuosiusIrMetodus(String kodas): void
        +analizuojamKintamaji(Kintamasis k, String kodas, Rysys r): void
        +analizuotiKintamojoPavadinima(Kintamasis k, String t, String p, Rysys r): void
        +gautiTipa(String p, boolean s): Tipas
        +analizuojamKompozicijosKintamaji(Kintamasis k, String kodas, Rysys r): void
        +analizuojamMetoda(String kodas): void
        +sudarytiIstisiniKoda(String adresas): String
        +salintiKasKabutese(String elute): String
        +atspausdintiKlasiuDiagrama(boolean visas): void
        +generuotiTeritorijas(): void
        +perskaiciuotiDydzius(String sr, int d): void
    }
    class Piesimui {
        ~lists: ArrayList<Klase>
        +sriftas: String
        +dydis: int
        +fonas: Color
        +spalva: Color
        +NurodytiKlasiuSarasa(ArrayList<Klase> k): void
        +Agregacija(int iki_x, int iki_y, int nuo_x, int nuo_y, Graphics g): void
        +Kompozicija(int iki_x, int iki_y, int nuo_x, int nuo_y, Graphics g): void
        +Paveldejimas(int nuo_x, int nuo_y, int iki_x, int iki_y, Graphics g): void
        +PiestiRysyi(Klase a, Graphics g): void
        +Piesti(Graphics g): void
    }
    Metodas "1" -- "*" Klase
    Tipas "1" -- "*" Klase
    Kintamasis "1" -- "*" Klase
    Klase "1" -- "*" Keturkampis
    Klase "1" -- "*" Rysys
    Klase "1" -- "*" Pradine
    Klase "1" -- "*" KlasiuDiagramosGeneravimas
    KlasiuDiagramosGeneravimas "1" -- "*" Pradine
    KlasiuDiagramosGeneravimas "1" -- "*" Piesimui
    
```

3. 5. 1 Klasijų detalizavimas

Klasė	Klasė
Apibrėžimas	Klasė skirta duomenims apie klasę saugoti
Atributai	Pavadinimas – klasės pavadinimas Kintamieji – klasės kintamieji Metodai – klasės metodai Rysiai – klasės ryšiai projektoKlase – ar klasė priklauso projektui.
Metodai	pridėtiKintamąjį – įtraukia nurodytą kintamąjį į klasės objektą pridėtiMetodą – įtraukia nurodytą metodą į klasės objektą pridėtiRysį – įtraukia nurodytą ryšį į klasės objektą perskaiciuotiTeritoriją – perskaiciuoja klasės užimamą teritoriją ekrane skaiciuotiMatmenis – skaičiuoja klasės keturkampio matmenis
Sąveikavimas	Ši klasė sąveikauja su klasėmis „Kintamasis“, „Metodas“, „Ryšys“, iš jų perimdama duomenis apie kintamuosius bei metodus.

35 lentelė. Klasės „Kintamasis“ detalizavimas

Klasė	Kintamasis
Apibrėžimas	Klasė skirta kintamojo duomenims atskirti
Atributai	Matomumas – kintamojo matomumas Statinis – ar kintamasis yra statinio tipo Pavadinimas – kintamojo pavadinimas Tipas – kokio tipo yra kintamasis Kiekis – kiek tokių kintamųjų yra Kompozicinis – ar kintamasis yra kompozicinis
Metodai	kintamojoFormatas() – suformuoja kintamojo eilutę.
Sąveikavimas	Ši klasė sąveikauja su klase „Tipas“, jai perduodama duomenis apie save.

36 lentelė. Klasės „Metodas“ detalizavimas

Klasė	Metodas
Apibrėžimas	Klasė skirta metodo duomenims atskirti
Atributai	Matomumas – metodo matomumas Statinis – ar metodas yra statinio tipo Pavadinimas – metodo pavadinimas grazinamasTipas – kokio tipo yra metodas Parametrai – metodo parametrų sąrašas
Metodai	metodoFormatas() – suformuoja metodo eilutę.
Sąveikavimas	Ši klasė sąveikauja su klase „Tipas“, jai perduodama duomenis apie save.

37 lentelė. Klasės „Piesimui“ detalizavimas

Klasė	Piesimui
Apibrėžimas	Klasė skirta piešimui vykdyti
Atributai	Šriftas – šriftą nusakantis kintamasis Dydis – šrifto dydį nusakantis kintamasis Fonas – fono spalvą nusakantis kintamasis Spalva – teksto spalvą nusakantis kintamasis
Metodai	nurodytiKlasiuSarasa(ArrayList<Klase> k) – metodas skirtas visam klasių sąrašui išsaugoti agregacija – piešia agregacijos rodyklę kompozicija – piešia kompozicijos rodyklę paveldėjimas – piešia paveldėjimo rodyklę PiestiRysi – piešia ryšį tarp dviejų klasių Piesti – piešia klasių diagramą
Sąveikavimas	-----

38 lentelė. Klasės „Keturkampis“ detalizavimas

Klasė	Keturkampis
Apibrėžimas	Klasė skirta vaizduojamam klasės keturkampiui nusakyti
Atributai	maxX – didžiausia x reikšmė x_nuo – nuo kokio taško pagal x ašį y_nuo – nuo kokio taško pagal y ašį aukštis – atstumas nuo keturkampio viršaus iki metodų srities pradžios aukštis2 – keturkampio aukštis plotis – keturkampio plotis
Metodai	kaistiPadeti – perkelia keturkampį į nurodytą vietą ieina – ar nurodytas taškas įeina į keturkampį
Sąveikavimas	Šią klasę paveldė „Klase“ klasė.

39 lentelė. Klasės „KlasiuDiagramosGeneravimas“ detalizavimas

Klasė	KlasiuDiagramosGeneravimas
Apibrėžimas	Klasė atliekanti pagrindinį duomenų surinkimo ir klasės objektų sudarymą
Atributai	projektoKlasiuAdresas – nurodytas sąrašas klasių adresų klasiuSarasas – klasių sąrašas tipuSarasas – tipų sąrašas dabartine – klasė su kuria dabar dirbama
Metodai	KlasiuDiagramosGeneravimas – klasės konstruktorius, kuris nustato pradinius kintamuosius gautiKlasiuSarasa – išgauna klasių sąrašą iš nurodytos direktorijos analizuotiVisaProjekta – pradeda projekto kodo analizavimą arKabutese – patikrina ar nurodyta eilutė yra kabutėse analizuotiFailoKoda – analizuoja vieno failo kodą analizuotiAntraste – analizuoja kodo antraštę gautiKlase – gražina klasės objektą su nurodytu pavadinimu analizuotiKintamuosiusIrMetodus – analizuoja klasės viduje esantį kodą analizuojamKintamaji – analizuoja nurodytą kintamąjį analizuotiKintamojoPavadinima – analizuoja nurodyto kintamojo pavadinimą gautiTipa – gražina tipo objektą su nurodytu pavadinimu analizuojamKompozicijosKintamaji – analizuoja kintamąjį, ar jis yra kompozicinis analizuojamMetoda – analizuoja metodo aprašymą sudarytiIstisiniKoda – paverčia visą kodą į vieną eilutę salintiKasKabutese – panaikina tekstą kabutėse generuotiTeritorijas – generuoja klasės teritoriją atvaizdavimui perskaiciuotiDydzius – perskaičiuoja klasių keturkampių dydžius
Sąveikavimas	Ši klasė sąveikauja su klase „Klasė“, jai perduodama visus duomenis apie klases.

40 lentelė. Klasės „Pradine“ detalizavimas

Klasė	Pradine
Apibrėžimas	Pagrindinė klasė iš kurios vykdomi tolesni veiksmai
Atributai	Bnd – klasių diagramos generavimo kintamasis Krastas – ar tai yra klasės keturkampio krašto riba Listas – klasių sąrašas Piesimui1 – piešimo kintamasis
Metodai	loadFile – „rkd“ failo užkrovimas saveFile – „rkd“ failo saugojimas
Sąveikavimas	Ši klasė sąveikauja su „klase“, „klasiuDiagramosGeneravimas“ ir „Piesimui“ klasėmis

41 lentelė. Klasės „Rsys“ detalizavimas

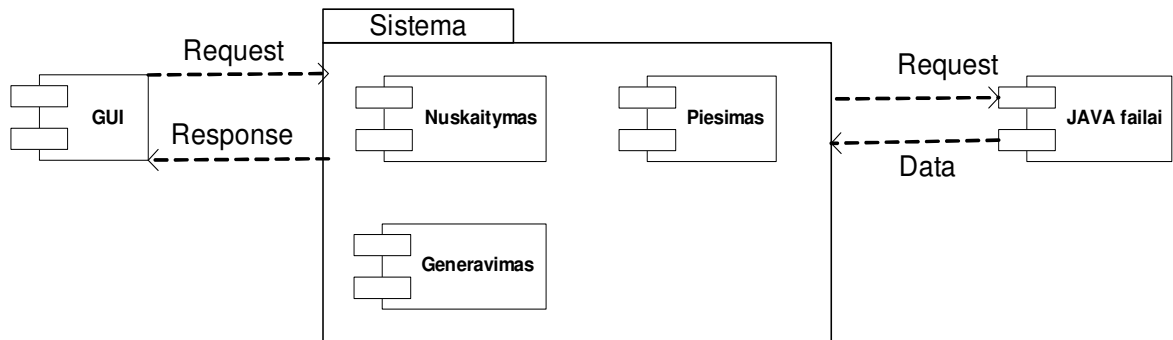
Klasė	Ryšys
Apibrėžimas	Klasė skirta ryšio aprašymui
Atributai	Tipas – ryšio tipui nusakyti Kardinalusmas – kardinalumui nusakyti Klase – su kokia klase ryšys rysysPer – per kokį kintamąjį yra ryšys
Metodai	-----
Sąveikavimas	Ši klasė sąveikauja su klase „Klasė“, jai perduodama visus duomenis apie klases.

42 lentelė. Klasės „Tipas“ detalizavimas

Klasė	Tipas
Apibrėžimas	Klasė skirta tipo aprašymui
Atributai	Klase – kokios klasės tipas Masyvas – ar daugiau nei viena klasė
Metodai	-----
Sąveikavimas	Ši klasė sąveikauja su klasėmis „Kintamasis“ ir „Metodas“, joms perduodama visus duomenis apie save.

3.6 Programinės įrangos komponentai

Pagrindiniai sistemos komponentai: GUI, pati sistema, ir JAVA failai. Vartotojo sąsaja „GUI“ pateikia užklausas sistemai. Sistema gali atlikti nuskaitymo, piešimo ir diagramos generavimo funkcijas, kuriems duomenis gauna iš „java“ failų (žr. 23 pav.).



23 pav. Sistemos komponentų tarpusavio sąveikos diagrama

3.7 Programinės įrangos kokybė

Produktas kuriamas naudojantis atviro kodo technologijomis, kas padidina išplečiamumo galimybes, kadangi vartotojai turi galimybę patys tobulinti ir redaguoti programą priklausomai nuo jų poreikių.

Kadangi programa kuriama JAVA programavimo kalba, o visos reikalingos bibliotekos integruotos į paketą, produktą galima laisvai perkelti į kitą kompiuterį ar operacinę sistemą.

4. TESTAVIMAS

4.1 Testavimo paskirtis

4.1.1 Testavimo tikslai

Aptikti galimas klaidas, jų priežastis, nustatyti sistemos funkcionalumą bei suderinamumą su kitomis sistemomis.

4.1.2 Testavimo resursai

- Nešiojamas kompiuteris Toshiba Satellite A200, kurio resursai:

Intel Core2 Duo 1,8Ghz procesorius;

2GB RAM;

Nvidia GeForce HD2400 512MB vaizdo korta;

Windows 7 operacinė sistema;

- Nešiojamas kompiuteris Dell Latitude D620, kurio resursai:

Intel Centrino Duo 1,8GHz procesorius;

4GB RAM;

Nvidia Quadro NVS 110M 64MB vaizdo korta;

Windows Vista SP2 operacinė sistema;

- Apple mini stacionarus kompiuteris, kurio resursai:

Intel Centrino 1,8Ghz procesorius

480MB RAM;

Windows XP SP2 operacinė sistema;

4.1.3 Pagrindiniai apribojimai

Kompiuteryje būtinai turi būti įdiegta Java palaikymo sistema, kitaip programos testavimas net neatliekamas.

4.2 Programinės įrangos funkcijų testavimas

43 lentelė. Failų atidarymo testavimas

Testo tikslas:		Testuojama failų atidarymo skiltis.				
Testo ID	Reikalavimai / tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
1	Nuskaityti JAVA projekto failai	Atidaroma direktorija su java failais	Ekrane pateikiami failai ir jų turinys	Ekrane pateikti failai ir jų turinys	T	
2		Atidaroma direktorija, kurioje nėra java failų	Ekrane niekas nepateikiama	Ekrane niekas nepateikiama	T	
3		Nubraižius vieną klasių diagramą norima braižyti naują, atidaromi kiti failai.	Braižoma nauja klasių diagrama.	Braižoma nauja klasių diagrama.	T	
4		Atidaromi java failai su klaidingu kodu viduje	Nebraižoma klasių diagrama	Nebraižoma klasių diagrama	T	

5	Nuskaitomas failas su plėtiniu „rkd”	Atidaromas rkd failas	Programa paruošta diagramos braižymui	Programa paruošta diagramos braižymui	N	Programa neinformuoja apie sėkmingą arba nesėkmingą failo atidarymą.
6		Atidaromas failas su plėtiniu rkd, bet jo turinys neatitinka laukiamo turinio	Neatliekami jokie veiksmai	Neatlikti jokie veiksmai		

44 lentelė. Failų išsaugojimo testavimas

Testo tikslas:		Testuojama failų išsaugojimo skiltis				
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
7	Diagramos išsaugojimas	Diagrama išsaugojama paveikslėlio formatu	Nurodytoje vietoje atsiranda paveikslėlis	Nurodytoje vietoje atsiranda paveikslėlis	T	
8		Diagrama išsaugojama rkd formatu	Nurodytoje vietoje atsiranda rkd failas	Nurodytoje vietoje atsiranda rkd failas	T	
9		Bandoma išsaugoti n nubraižytą diagramą rkd formatu	Veiksmas negalimas	Išsaugojama tuščia diagrama	N	Reikėtų neleisti atlikti tokios funkcijos.
10		Bandoma išsaugoti nubraižytą diagramą, kurioje pašalintos visos klasės	Išsaugojama tuščia diagrama	Išsaugojama tuščia diagrama	T	
11		Bandoma išsaugojant nutrinti “.rkd” priedą.	Veiksmas negalimas	Išsaugojamas failas be plėtinio	N	Programa išsaugoja failą be jokio plėtinio.
12		Bandoma išsaugojant nutrinti “.png” priedą.	Veiksmas negalimas	Išsaugojamas failas be plėtinio	N	

45 lentelė. Klasių diagramos braižymo testavimas

Testo tikslas:		Testuojama klasių diagramos braižymo skiltis				
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
13	Diagramos braižymas	Atidarius failus braižoma diagrama	Nubraižoma klasių diagrama	Nubraižoma klasių diagrama	T	
14		Atidaromas išeities kodas, kuriame yra 5 klasės	Diagramoje matomos visos 5 klasės	Diagramoje matomos visos 5 klasės	T	
15		Atidaromas išeities kodas, kuriame yra 8 klasės	Diagramoje matomos visos 8 klasės	Diagramoje matomos visos 8 klasės	T	
16		Atidaromas išeities kodas, kuriame yra 11 klasių	Diagramoje matomos visos 11 klasių	Diagramoje viena klasė yra dalinai už ekrano ribų	T	Matomos visos klasės, sudėtingas jų išdėstymas

17		Atidaromas išeities kodas, kuriame yra 18 klasių	Diagramoje matomos visos 18 klasių	Diagramoje matomos tik 12 klasių	N	Reikia koreguoti atvaizdavimą kodą, kad būtų atvaizduojamos visos klasės.
----	--	--	------------------------------------	----------------------------------	---	---

46 lentelė. Klasės duomenų generavimo testavimas

Testo tikslas:		Testuojamas klasės duomenų išgavimą iš Java šaltinio kodo			
Testo ID	Reikalavimai/tiksiai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N
18	Klasių diagramos struktūros sudarymo teisingumas	Pateikiamas vienas java failas su viena aprašyta klase be klasės kintamųjų ir metodų	Atvaizduojama viena tuščia klasė su jos pavadinimu	Failas Keisti Main	T
19		Pateikiamas vienas java failas su dviejų klasių aprašais, kurios neturi klasės kintamųjų ir metodų	Atvaizduojamos dvi tuščios klasės su savo pavadinimais	Failas Keisti Main Kita	T
20		Pateikiamas vienas java failas su viena aprašyta klase su klasės kintamaisiais ir metodais	Atvaizduojama viena klasė su savo pavadinimu, kintamaisiais ir metodais	Failas Keisti Main - bin: boolean - vardas: String + amzius: int + Main(): void + start(): void	T
21		Pateikiami keturi java failai, turintys tarpusavio paveldėjimo ir agregacijos/ kompozicijos ryšių, o viename faile aprašant dvi klases (žr. 2 priedas.)	Sugeneruojama klasių diagrama iš 5 klasių: Klasiuke paveldi Irasas klasę Irasas paveldi Lngas klasę Main klase siejasi su Klasiuje ir Lngas klasėmis agregacijos ryšiu Mano klase nesisieja su kitomis klasėmis	Sugeneruojama klasių diagrama iš 5 klasių: Klasiuke paveldi Irasas klasę Irasas paveldi Lngas klasę Main klase siejasi su Klasiuje ir Lngas klasėmis agregacijos rysių Mano klase nesisieja su kitomis klasėmis (žr. 2 priedas. 47 pav.)	T

22		Pateikiamas vienas java failas su viena aprašyta klase be klasės kintamųjų ir metodų, bet turintis vidinės klasės aprašą	Atvaizduojama diagrama su dviem klasėmis	Failas Keisti Main BND + sk: double - kascia(): void	T
23		Pateikiami trys java failai, turintys klasių su vidinėmis aprašytomis klasėmis (bent 2 lygių vidinės klasės)	Atvaizduojamos visos aprašytos klasės, ir pagrindinės ir vidinės	Failas Keisti BND2 + sk: int Dar ~ aks: double Main ~ bnd2: BND ~ bnd: BND2 + main(String[] args): void + kazkas(): void + kazkas2(): void BND + sk: double - kascia(): void vidinesVidine + skaicukas: int	T

47 lentelė. Klasės duomenų atvaizdavimo testavimas

Testo tikslas:		Testuojamas klasės duomenų atvaizdavimas				
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
24	Atvaizduojami klasių duomenys esant dideliame procesoriaus apkrovimui	Parinkta atvaizduoti duomenys iš ypatingai didelių java failų	Pateikiami visi duomenys	Pateikiami visi duomenys	T	Nepastebėta atvaizdavimo užtrukimo
25		Parinkta atvaizduoti 40 klasių duomenys	Pateikiami visi duomenys	Pateikiami visi duomenys	T	Nepastebėta atvaizdavimo užtrukimo.

4.3 Programinės įrangos charakteristikų testavimas

4.3.1 Našumo testavimas

48 lentelė. Failų atidarymo greičio testavimas

Testo tikslas:		Testuojamas failų atidarymo greitis				
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
26	Nuskaityti failai esant ypatingai dideliame procesoriaus apkrovimui	Atidaroma aplanka su java failais	Ekrane pateikiama sugeneruota klasių diagrama ne ilgiau nei per 0,5 s	Ekrane pateikta sugeneruota klasių diagrama per 0,3 s	T	Pastebėti nežymūs failų atidarymo užtrukimai tik tada kai nurodomas didesnis kelias iki failų.

27		Atidarymui nurodomas visas C diskas	Jeigu yra java failų jie atvaizduojami ekrane	Teko laukti ilgiau nei 5 minutes todėl programa buvo nutraukta.	N	Labai ilgai užtruko failų paieška.
28		Atidaromi maždaug po 3000 eilučių turintys java failai	Ekrane pateikiama sugeneruota klasių diagrama ne ilgiau nei per 1s	Ekrane pateikta sugeneruota klasių diagrama per 0,7 s	T	Pateikus didelio turinio failus programa nežymiai užtruko juos nuskaitant
29		Atidarymui parenkami 5 java failai	Ekrane pateikiama klasių diagrama ne ilgiau nei per 0,5 s	Ekrane pateikta klasių diagrama per 0,3 s	T	Programa tai atliko mažiau nei per sekundę.
30		Atidarymui parenkama 15 java failų	Ekrane pateikiama klasių diagrama ne ilgiau nei per 1 s	Ekrane pateikiama klasių diagrama per 0,5 s	T	Programa tai atliko mažiau nei per sekundę

49 lentelė. Diagramos braižymo stresinis testavimas

Testo tikslas:		Testuojamas diagramos braižymo esant didelei apkrovai				
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
31	Atvaizduojamos diagramos esant dideliui procesoriaus apkrovimui	Parinkta atvaizduoti diagrama iš daugiau nei 3000 eilučių turinčių failų	Pateikiama klasių diagrama	Ekrane matomos tik kelios klasės	N	Klasės turinčios daug kintamųjų bei metodų užima labai didelę dalį ekrane ir nebesimato kitų klasių, o atvaizdavimo laikas nesulėtėjęs.
32		Parinkta atvaizduoti iš 40 klasių	Klasės atvaizduojamos ekrane	Ekrane matomos ne visos klasės	N	Neatvaizduojamos visos klasės, bet atvaizdavimo laikas nesudaro nepatogumų.

4. 3. 2 Suderinamumo testavimas

50 lentelė. Programos jautrumo ekrano rezoliucijai testavimas

Testo tikslas:		Testuojamas programos jautrumas ekrano rezoliucijai				
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
33	Dirbama su programa prie 1280x800 rezoliucijos	Paleista programa	Programos langas neiškraipytas ir viskas aiškiai matyti	Programos langas neiškraipytas ir viskas aiškiai matyti	T	

34		Atvaizduota klasių diagrama	Klasių diagrama atvaizduota ir viskas matoma.	Klasių diagrama atvaizduota ir viskas matoma.	T	
35	Dirbama su programa prie 800x600 rezoliucijos	Paleista programa	Programos langas neiškraipytas ir viskas aiškiai matyti	Normalaus dydžio (esant 1280x800 raiškai) programos langas užima daugiau ekrano vietos	T	
36		Atvaizduota klasių diagrama	Klasių diagrama atvaizduota ir viskas matoma	Normalaus dydžio (esant 1280x800 raiškai) programos langas užima daugiau ekrano vietos	T	

4. 3. 3 Atstatymo testavimas

51 lentelė. Sistemos atstatymo testavimas

Nr.	Situacija	Sprendimas
37	Atidarius programą parinkta nuskaitymui visas kietasis diskas ir paleistas nuskaitymas	Programa visiškai nepakimba, bet labai ilgai dirba savo darbą, vartotojas tai gali nutraukti tiesiog išjungęs programą, aišku pats išjungimo procesas gali užtrukti

4. 4 Testavimo išvados

Testavimo metu gauti rezultatai įrodo jog programa yra tinkama naudoti ir vartotojui nesukels didelių nepatogumų. Tokias klaidas kaip klasių netilpimas į ekraną būtų galima ištaisyti ateityje.

5. VARTOTOJO DOKUMENTACIJA

5.1 Vartotojo vadovo paskirtis

Šis vartotojo vadovas skirtas supažindinti su programa ir išmokinti vartotoją elgtis su ja.

5.2 Sistemos funkcinis aprašymas

Klasių diagramų braižymo iš programos išeities teksto programa atlieka pagrindines tam tikslui reikalingas funkcijas. Programai perduodami failai su išeities kodais ir gaunamas rezultatas – klasių diagrama. Jei programos sukurtas automatinis klasių išdėstymas netinkamas, vartotojas gali tai padaryti pats.

5.3 Sistemos vartotojo vadovas

5.3.1 Pagrindinis programos langas

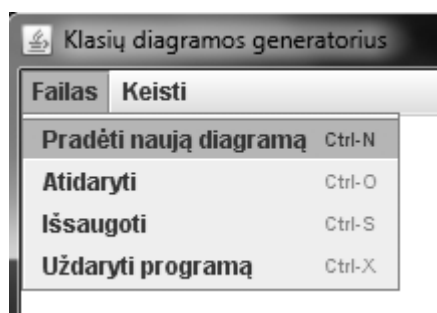


24 pav. Pagrindinis programos langas

Pagrindiniame programos lange (žr. 24 pav.) yra atvaizduojama klasių diagrama, taip pat iš šio lango galima pasiekti ir kitas programos funkcijas.

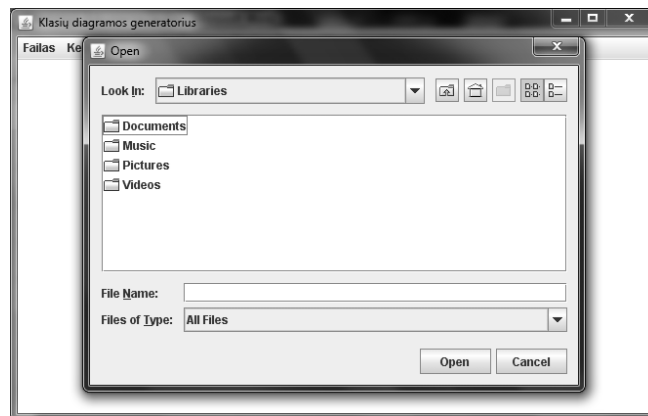
5.3.2 Klasių diagramos kūrimas

- Failų išrinkimas

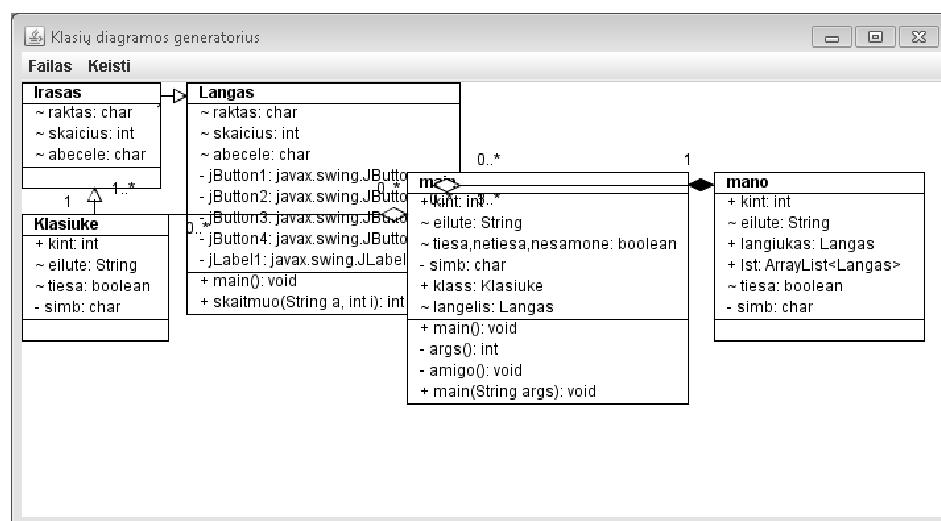


25 pav. Naujos diagramos kūrimas

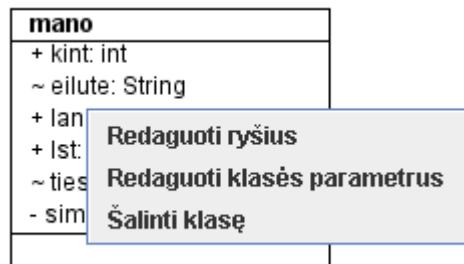
Pirmas veiksmas kurį turite padaryti norint pradėti klasių diagramos braižymą – turite nurodyti programai jūsų projekto katalogą, kuriame turi būti „java“ failai. Tai galite padaryti iš meniu punkto „Failai“ išrinkę papunktį „Pradėti naują diagramą“ (žr. 25 pav.). Pasirinkę šį punktą išvysite katalogo pasirinkimo langą (žr. 26 pav.), kuriame turėsite nurodyti savo projekto katalogą su „java“ failais.



Atlikus šiuos veiksmus ir jeigu jūsų pateiktų „java“ failų kodai yra parašyti teisingai, ekrane išvysite automatiškai sugeneruotą klasių diagramą (žr. 27 pav.). Jos vaizdas iš karto gali būti nepatrauklus, todėl jums teks rankiniu būdu poredaguoti iki norimo galutinio varianto.



28 pav. Klasės antraštė



29 pav. Išskleidžiamas klasės meniu

Tai padarius atsiveria naujas langas, kuriame matomi išdėstyti viršutinėje dalyje klasės atributai, o apatinėje – metodai (žr. 30 pav.). Tiek atributai, tiek ir metodai atvaizduojami lentelės pagalba, tam kad būtų galima lengvai redaguoti kiekvieną parametą.

The image shows a dialog box titled 'Klasės parametru redagavimas'. It has two tabs: 'Pridėti' and 'Ištrinti'. The 'Pridėti' tab is active. It contains a text field for 'Klasės pavadinimas:' with the value 'main' and a button 'Keisti klasės pavadinimą'. Below this is a table with columns: 'Matomumas Str', 'Statinis', 'Pavadinimas', 'Tipas Str', and 'Kiekis'. The table contains the following data:

Matomumas Str	Statinis	Pavadinimas	Tipas Str	Kiekis
public	☒	kint	int	0
package	☒	eilute	String	0
package	☐	tiesa,netiesia,...	boolean	0
private	☐	simb	char	0
public	☐	klass	Klasiuke	0
package	☐	langelis	Langas	0

Below the table is a button 'Redaguoti kintamąjį'. Below that is another table with columns: 'Pavadinimas', 'Statinis', 'Matomumas Str', 'Parametrai Str', and 'Grazinamas'. The table contains the following data:

Pavadinimas	Statinis	Matomumas Str	Parametrai Str	Grazinamas
main	☒	public		
args	☒	private		int
amigo	☒	private		
main	☒	public	String args	

Below the table is a button 'Redaguoti metodą'.

30 pav. Klasės parametru redagavimas

- Atributų galimi parametrai:
 - Matomumas – tai atributo matomumas, kuris gali būti public, private, protected arba package.
 - Statinis – šis laukas nusako ar atributas yra statinis.
 - Pavadinimas – atributo pavadinimas.
 - Tipas – atributo tipas.
- Metodų galimi parametrai:
 - Pavadinimas – metodo pavadinimas

- Statinis – nusako ar metodas yra statinis
- Matomumas – taip pat kaip ir atributo, gali būti public, private, protected arba package.
- Parametrai – nurodomi metodo parametrai.
- Gražinamas – tai metodo gražinamų duomenų tipas (tuščia reiškia nieko negražinama)

Lentelių laukų redagavimas atliekamas du kartus spustelėjus kairį pelės klavišą ant norimo lauko, atlikus pakeitimus pereinama į kitą lauką arba spaudžiamas „Enter“ klavišas.

Lango meniu juostoje matomi du punktai: „Pridėti“ ir „Ištrinti“ (žr. 30 pav.). Šių punktų pagalba galima pridėti arba pašalinti norimą kintamąjį arba metodą iš klasės parametrų sąrašo.

- Naujo klasės atributo įtraukimas.

Norint pridėti naują atributą išrenkamas meniu punktas „Pridėti“ ir pasirenkama „Kintamasis“. Atsiveria naujas langas (žr. 31 pav.), kuriame reikia suvesti naujo kintamojo duomenis.

31 pav. Naujo parametro kūrimas

Suvedus šiuos duomenis spaudžiama „Gerai“ ir duomenys įtraukti į klasės atributų sąrašą.

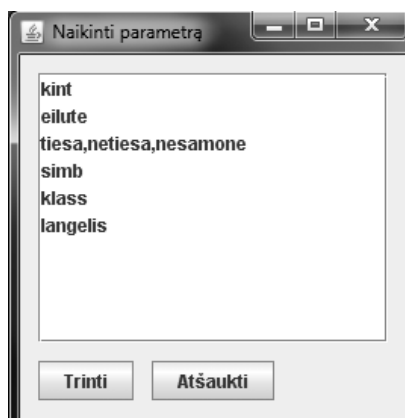
- Naujo klasės metodo įtraukimas.

Naujas klasės metodas įtraukiamas tokiu pat būdu kaip ir kintamasis, tik meniu punkte „Pridėti“ išrenkamas „Metodą“ punktas ir užpildomi metodo parametrus atitinkantys laukai (žr. 32 pav.).

32 pav. Naujo metodo kūrimas

- Klasės atributų šalinimas.

Norint ištrinti klasės atributą, meniu punkte „Ištrinti“ pasirenkame „Kintamąjį“, Atsivėrusiame lange (žr. 33 pav.) išrenkamas norimas atributas arba keli atributai ir spaudžiamas „Trinti“ mygtukas.



33 pav. Klasės atributų trynimas

- Klasės metodų šalinimas.

Norint ištrinti klasės metodą, atliekami tokie pat veiksmai kaip ir trinant klasės kintamąjį, tik meniu punkte „Ištrinti“ pasirenkamas „Metodą“ punktas ir atsivėrusiame lange išrenkamas norimas metodas/metodai ir spaudžiamas patvirtinimo mygtukas.

- Klasės ryšių redagavimas

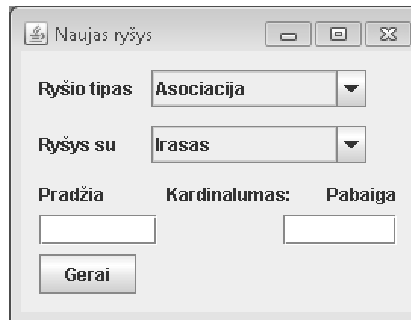
Norint redaguoti klasės ryšius reikia ant norimos klasės spausti dešinį pelės klavišą ir parinkti meniu punktą „Ryšių redagavimas“. Tokiu būdu atveriamas naujas langas (žr. 34 pav.), kurio pagalba bus galima atlikti veiksmus su ryšiais.



34 pav. Ryšių redagavimas

- Naujo ryšio įtraukimas.

Norint įtraukti naują ryšį parenkamas meniu punktas „Naujas ryšys“. Tokiu būdu atveriamas langas su naujo ryšio parametrų parinkimu (žr. 35 pav.). Pasirinkus norimo ryšio parametrus spaudžiamas patvirtinimo mygtukas ir ryšys užregistruojamas.



35 pav. Naujas ryšys

- Ryšio šalinimas.

Norint pašalinti ryšį iš sąrašo reikia atsistoti lentelėje ant norimo ištrinti ryšio ir spaudžiamas „Naikinti ryšį“ mygtukas.

- Naujos klasės kūrimas bei netinkamos šalinimas

Norint sukurti naują klasę diagramoje, reikia tuščioje vietoje spustelėti dešinį pelės klavišą ir spausti nat „Nauja klasė“ (žr. 36 pav.). Bus sukurta nauja klasė, kurios parametrus bei pavadinimą bus galima redaguoti kaip ir kitų klasių.



36 pav. Meniu punktas „Nauja klasė“

Norint pašalinti netinkamą klasę reikia ant jos spustelėti dešinį pelės klavišą ir pasirinkti meniu punktą „Šalinti klasę“. Tokiu atveju bus pašalinta klasė ir visi su ja susiję ryšiai.

- Diagramos vaizdo redagavimas

Programoje galima redaguoti bendrą diagramos vaizdą, t.y.: diagramos fono spalvą, teksto šriftą, dydį bei jo spalvą. Tai galima atlikti pasirinkus meniu punktą „Keisti“ -> „Diagramos vaizdas“. Tokiu būdu atveriamas naujas langas (žr. 37 pav.), kuriame galima keisti anksčiau išvardintus parametrus.



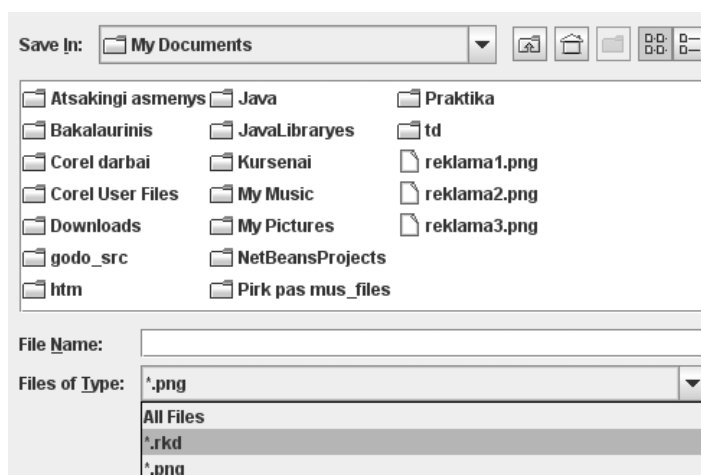
37 pav. Diagramos vaizdo keitimas

- Išsaugojimas

Sukurtą klasių diagramą galima išsaugoti dviem būdais. Pirmasis būdas naudingas tuo atveju jeigu jūs norite ateityje turėti galimybę redaguoti sukurta klasių diagramą. Norint išsaugoti šiuo būdu reikia pasirinkti meniu punktą „Failas“ -> „Išsaugoti“ ir atsivėrusiame lange pasirinkti „rkd“ failo

formatą (žr. 38 pav.). Taip yra išsaugojama visa diagramos informacija tolimesniam naudojimui. Išsaugoto failo plėtinys turi būti „rkd“. Tokius failus galima atidaryti pasirinkus meniu punktą „Failas“ -> „Atidaryti“ . Atidarius „rkd“ failą, diagrama bus matoma tokia pat kaip ir prieš išsaugant ir su ja bus galima atlikti ankstesniame skyriuje aptartus veiksmus.

Antrasis saugojimo būdas yra naudojamas diagramos peržiūrai, demonstracijoms ar kitokiam vaizdiniam panaudojimui. Diagrama išsaugoma paveikslėlio formatu su plėtiniu „png“. Norint išsaugoti diagramą kaip paveikslėlį reikia pasirinkti meniu punktą „Failas“ -> „Išsaugoti“ ir atsivėrusiame lange pasirinkti „png“ failo formatą.



38 pav. Failo saugojimas

5.4 Sistemos diegimo vadovas

Programa neturi specialaus diegimo vedlio, tiesiog programos failą reikia įkelti į norimą vietą kietajame diske ir jį paleisti.

IŠVADOS

Atlikus klasių diagramos kūrimui galimų naudoti programų analizę pastebėta, kad tokio tipo sistemos yra sudėtinės, turinčios daug papildomų funkcijų, kurios gali būti ir nereikalingos klasių diagramų kūrimui. Visos išnagrinėtos programos skiriasi viena nuo kitos: vienos neturi vienokių funkcijų, kitos kitokių. Be to dauguma tokio tipo programų palaiko tik anglų kalbą, tad galima daryti prielaidą, kad konkrečiai Lietuvos rinkai skirtų tik klasių diagramų braižymo iš Java šaltinio kodų sistemų stinga.

Pagal tokio tipo programoms keliamus reikalavimus parengtos reikalavimų ir architektūros specifikacijos, kuriose naudojamos UML diagramos ir jų detalūs aprašai. Tad visus programai keliamus reikalavimus galės nesunkiai perprasti tiek UML diagramų notaciją išmanantys, tiek ir jos nežinantys vartotojai.

Programa realizuota Java programavimo kalba, todėl gali būti lengvai įdiegta ir naudojama beveik visose operacinėse sistemose.

Realizuotos visos pagrindinės funkcijos, reikalingos tokio tipo programai, t.y. klasių diagramos generavimas iš kodo, jos redagavimas, naujų klasių kūrimas bei netinkamų šalinimas, išsaugojimas paveikslėlio bei vėliau galimo redaguoti failų formatais. Tad programa gali būti taikoma praktikoje ir pilnai tenkinti visus su klasių diagramomis dirbančių asmenų poreikius.

Atliktas išsamus testavimas ir nustatyta, kad programa tinkama naudojimui ir užtikrina teisingą klasių diagramos generavimą bei neturi kritinių klaidų, todėl jos vartotojams neturėtų sukelti naudojimo nepatogumų.

Jei vartotojas nemokėtų naudotis šia programa, tai jis galėtų pasinaudoti vartotojo vadovu, kuriame aprašytos visos programos funkcijos bei jų atlikimo eiga, tad jokių nesklandumų toliau dirbant su programa nebeturėtų kilti.

LITERATŪRA

1. Digitron Software, *Class Diagrams - Quickly and Easily* [interaktyvus], [žiūrėta 2010-05-21]. Prieiga per internetą <<http://www.classdraw.com>>;
2. Donald Bell, *UML basics: The class diagram, An introduction to structure diagrams in UML 2* [interaktyvus], 2004-09-15 [žiūrėta 2010-05-01]. Prieiga per internetą <<http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell>>;
3. Microsoft Office Visio 2007 product overview [interaktyvus], 2010 [žiūrėta 2010-05-01]. Prieiga per internetą <<http://office.microsoft.com/en-us/visio/HA101656401033.aspx>>;
4. No Magic, Inc, *Introducing MagicDraw* [interaktyvus], 2010 [žiūrėta 2010-05-01]. Prieiga per internetą <http://www.magicdraw.com/what_is>.
5. Oracle Corporation, *Unified Modeling Language (UML)* [interaktyvus], 2010 [žiūrėta 2010-05-01]. Prieiga per internetą <<http://netbeans.org/features/uml>>;
6. Sacha Barber, *100% Reflective Class Diagram Creation Tool* [interaktyvus], 2007 [žiūrėta 2010-05-01]. Prieiga per internetą <<http://www.codeproject.com/KB/cs/AutoDiagrammer.aspx>>;
7. Scott W. Ambler, *UML 2 Class Diagrams* [interaktyvus], 2009 [žiūrėta 2010-05-01]. Prieiga per internetą <<http://www.agilemodeling.com/artifacts/classDiagram.htm>>;
8. The Eclipse Foundation, *Project Plan For Model Development Tools (MDT), version Helios* [interaktyvus], 2010 [žiūrėta 2010-05-01]. Prieiga per internetą <<http://www.eclipse.org/projects/project-plan.php?projectid=modeling.mdt>>;
9. Tom Gooch, *What is UML?* [interaktyvus], [žiūrėta 2010-05-21]. Prieiga per internetą <http://atlas.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/what_is_uml.htm>;

TERMINŲ IR SANTRUMPŲ ŽODYNĖLIS

Failas	- duomenų rinkinys, turintis vardą.
JAVA	- objektiškai orientuota programavimo kalba.
Java Virtuali Mašina	- rinkinys kompiuterinių programų ir duomenų struktūrų, kurios naudojamos virtualios mašinos modeliui norint paleisti kitas kompiuterines programas arba jų sudedamąsias dalis.
PNG Failas	- grafikos failas, suglaudintas fotografijai pritaikytu diskrečiuoju kosinusų transformacijos metodu.
Klasė	- aprašas, nusakantis tam tikro tipo objektų duomenų struktūrą ir manipuliavimo jais taisykles (elgesį).
Klasių diagrama	- statinės struktūros diagrama, aprašanti sistemos klases bei ryšius tarp jų.
Klasių diagramų notacija	- nurodymų kaip braižomos klasių diagramos rinkinys.
Programos išeities kodas	- bet kokia sakinių seka, užrašyta žmogui suprantama programavimo kalba.
Redaguojamas failas	- failas, kurį atidarius su tam tikra programa galima redaguoti jo turinį.
Ryšys tarp klasių	- apibudina santykį tarp klasių (paveldėjimas, kompozicija, agregacija, asociacija).
UML	(<i>ang. Unified Modeling Language</i> , Vieninga modeliavimo kalba) - Objektiškai orientuotų programų dokumentų modeliavimui ir specifikacijų kūrimui yra sukurta speciali kalba.