

**VILNIAUS UNIVERSITETAS  
KAUNO HUMANITARINIS FAKULTETAS**

**INFORMATIKOS KATEDRA**

Verslo informatikos studijų programa  
**Kodas 62109P101**

Tadas Lingė

**MAGISTRO BAIGIAMASIS DARBAS**

**„MOBILIŲ ĮRENGINIŲ PROGRAMAVIMO PRIEMONIŲ GALIMYBIŲ  
ANALIZĖ“**

Kaunas 2009

**VILNIAUS UNIVERSITETAS  
KAUNO HUMANITARINIS FAKULTETAS**

**INFORMATIKOS KATEDRA**

TADAS LINGĖ

**MAGISTRO BAIGIAMASIS DARBAS**

**„MOBILIŲ ĮRENGINIŲ PROGRAMAVIMO PRIEMONIŲ GALIMYBIŲ  
ANALIZĖ“**

Leidžiama ginti \_\_\_\_\_

Magistrantas \_\_\_\_\_  
(parašas)

Darbo vadovas: \_\_\_\_\_  
(parašas)

**Docentas, Dr Vytautas Evaldas Rudžionis** \_\_\_\_\_  
(darbo vadovo mokslo laipsnis, mokslo pedagoginis  
vardas, vardas ir pavardė)

Darbo įteikimo data \_\_\_\_\_

Registracijos Nr. \_\_\_\_\_

Kaunas 2009

# Turinys

PAVEIKSLŲ SĄRAŠAS.....	4
LENTELIŲ SĄRAŠAS.....	5
SUTRUMPINIMŲ SĄRAŠAS.....	6
SANTRAUKA.....	7
ĮVADAS.....	8
1. PROGRAMAVIMO PRIEMONĖS.....	10
1.1. Java ME programavimo aplinka.....	11
1.2. Eclipse IDE.....	14
1.3. NetBeans IDE.....	19
1.4. Python for S60.....	22
1.5. IntelliJ IDEA.....	24
1.6. Programinių įrankių palyginimas.....	27
2. PROJEKTINĖ DALIS.....	29
2.1. Projekto tikslas.....	29
2.2. Kompiuterizuojamų funkcijų charakteristika.....	30
2.3. Duomenų bazės aprašymas.....	31
2.4. Priemonės naudotos programinės įrangos kūrimui:.....	32
2.5. Programinės įrangos aprašymas.....	33
3. UNIVERSALIOS SĄSAJOS APLIKACIJOS KŪRIMO EKSPERIMENTINIS TYRIMAS.....	35
3.1. Eksperimentinis tyrimas.....	36
3.3. Pasiūlymai.....	43
IŠVADOS.....	45
LITERATŪRA.....	46
PRIEDAI.....	48
Taikomosios aplikacijos schema.....	48
Vartotojo vadovas.....	49
Pranešimo medžiaga. Universalios sąsajos aplikacijų kūrimas mobiliems įrenginiams.....	53

## PAVEIKSLŲ SĄRAŠAS

1 pav. Java platforma. ....	12
2 pav. CLDC platforma. ....	12
3 pav. Connected Device Profile (CDC).....	13
4 pav. Eclipse platformos vartotojo sąsaja. ....	14
5 pav. Eclipse platformos architektūra. ....	15
6 pav. NetBeans Mobility (JavaME) programos langas. ....	21
7 pav. Python kodo pavyzdžiai. ....	22
8 pav. IntelliJ IDEA programos langas. ....	24
9 pav. Informacinės sistemos funkcijų hierarchijos diagrama. ....	30
10 pav. Konceptinis duomenų bazės modelis. ....	31
11 pav. Programinės įrangos struktūra. ....	33
12 pav. Veiklos diagramos elementai. ....	44

## LENTELIŲ SĄRAŠAS

1 lentelė. IntelliJ IDEA licenzijų kainos .....	26
2 lentelė. Programavimo įrankių palyginimas.....	27
3 lentelė. Taikomosios aplikacijos testavimo rezultatai.....	37
4 lentelė. Taikomosios aplikacijos testavimo rezultatai.....	39
5 lentelė. Taikomosios aplikacijos testavimo rezultatai.....	41

## SUTRUMPINIMŲ SĄRAŠAS

1. IS – informacinė sistema.
2. IT – informacinės technologijos.
3. OS – operacinė sistema.
4. Psl. – puslapis.
5. Pav. – paveikslas.
6. Žr. – žiūrėti.
7. API – aplikacijų programavimo sąsaja.
8. IDE – Integruota programų kūrimo aplinka.
9. GUI – grafinė vartotojo sąsaja.
10. OC – operacinis kontekstas, apibūdina techninę ir programinę aplinką.

# SANTRAUKA

Tadas Lingė. (2009). *Analysis of Tools for Mobile Devices Programming*. Master's Work in Business Informatics. Graduation Paper. Vilnius University, Kaunas Faculty of Humanities, Department of Informatics. 45 p.

## SUMMARY

The goals of this master's work are analyzes tools for mobile devices programming and universal interface for different models creation opportunities. Also the paper analyzes Java ME, Visual Studio .NET, Python for S60, NetBeans IDE and BREW programming tools. Describe the features of the functioning of the programming tools, to set the programming options, listed the advantages and disadvantages. The best in this analyze was NetBeans IDE. There are created IS conceptual model of driving school and project of mobile software. According to the programming tools analyzes results and IS project was created new application. It is dedicated to improve driving school processes and make scientific research. Application was tested in different mobile platforms. The test shown application functions which works wrong. Most important functions fixed, but some of them can't be fixed, such as displaying full table of content. This bug can be fixed only displaying table horizontally. This experimental work shows some methods of universal application creation. Also there is suggested how we can create universal application by using UML diagrams.

## IVADAS

Magistro baigiamasis darbas atliekamas siekiant išsiaiškinti mobiliųjų įrenginių programavimo priemonių galimybes, jų panaudojimo galimybes bei universalios sąsajos sukūrimo galimybes. Darbe aprašytos mobiliųjų įrenginių programavimo priemonių galimybės, taikomosios aplikacijos projektavimas, pateikiami testavimo duomenys bei atliktas universalios sąsajos aplikacijos kūrimo eksperimentinis tyrimas.

**Temos aktualumas.** Mobilųjų įrenginių programavimo priemonėmis galima sukurti įvairių taikomųjų aplikacijų, tačiau dauguma jų veikia tik tam tikruose įrenginiuose. Todėl atsirado poreikis nustatyti universalios sąsajos kūrimo principus ir metodus bei sukurti universalios sąsajos taikomąją aplikaciją. Vairavimo mokyklos instruktoriai šia aplikacija naudosis praktiškai.

**Baigiamojo darbo problema** – rinkoje esančios mobiliųjų įrenginių programavimo priemonės, jų galimybių lyginimas bei praktinis pritaikomumas.

**Baigiamojo darbo objektas** – mobiliųjų įrenginių programavimo priemonės ir mobilūs įrenginiai.

**Baigiamojo darbo tikslas** – ištirti mobiliųjų įrenginių programavimo priemonių ir universalios sąsajos skirtingiems modeliams sukūrimo galimybes.

### **Darbo uždaviniai:**

- Išanalizuoti programavimo priemonių galimybes, privalumus ir trūkumus.
- Suprojektuoti vairavimo mokyklos informacinės sistemos taikomąją aplikaciją skirtą mobiliems įrenginiams
- Sukurti taikomąją aplikaciją vienu iš analizuotų mobiliųjų įrenginių programavimo įrankiu.
- Atlikti taikomosios aplikacijos testavimą.
- Atlikti universalios sąsajos kūrimo eksperimentinį tyrimą.

### Darbe naudojama **metodika:**

- Mokslinės literatūros analizė
- Techninių šaltinių analizė
- Palyginimo ir sintezės metodai
- Apibendrinimo metodas
- Pažintinio apėjimo metodas



**Darbo struktūra.** Šis baigiamasis darbas susideda iš trijų dalių: pirmoje dalyje atliekama programavimo priemonių galimybių analizė, antroje dalyje aprašomas taikomosios aplikacijos kūrimo projektas, o trečiojoje pateikiamas universalios sąsajos aplikacijos kūrimo eksperimentinis tyrimas. Darbo apimtis neįskaitant literatūros sąrašo ir priedų yra 45 lapai, darbe pateikta 12 paveikslų ir 4 lentelės.

**Darbo aprobacija.** Darbo rezultatai buvo pateikti ir aptarti 14-oje tarpuniversitetinėje magistrantų ir doktorantų mokslinėje konferencijoje „Informacinės technologijos“ (Kaunas, Vilniaus universitetas, Kauno humanitarinis fakultetas, 2009 m. gegužė 8 d.). Pranešimas įtrauktas į atitinkamų metų konferencijos pranešimų medžiagą.

# 1. PROGRAMAVIMO PRIEMONĖS

Mobilieji telefonai gali būti naudojami ne tik skambinti bei trumposioms žinutėms siųsti, bet ir keistis įvairiausiais duomenimis, siųsti kompiuterinėms programoms, naršyti internete, naudotis įvairiomis elektroninėmis paslaugomis. Dauguma interneto vartotojų nori šiomis galimybėmis naudotis mobilioje aplinkoje, nepriklausomai nuo buvimo vietos ir laiko.

Atsirado naujos galimybės rinkoje pasirodžius išmaniesiems telefonams, delniniams kompiuteriams bei jų programavimo priemonėms: NetBeans, Eclipse, IntelliJ IDEA, Python for S60 ir daugelis kitų. Šiais įrankiais pagaminta programinė įranga yra skirta mobiliems prietaisams, kurie paneigė visus iki tol įmanomus apribojimus. Įvairūs žaidimai, žodynai, žemėlapiai, kelionių gidai, MPEG grotuvai ir kt. – visa tai šiandien gali turėti kiekvienas mobiliojo ryšio vartotojas, turintis tobulesnį telefoną.

Java IDE (Integruota programų kūrimo aplinka) rinkoje yra tik trys pagrindiniai konkurentai, tačiau konkurencija tarp jų ypač intensyvi. NetBeans ir Eclipse yra nemokami ir atviro kodo paketai, bei apima abi Java ME platformos konfigūracijas, tuo tarpu IntelliJ yra mokamas programavimo įrankis. Java ME platformos programavimo įrankiams konkurenciją sudaro C/C++ programavimo įrankis Python for S60. Jis yra unikalus, nes veikia pačiame mobiliame įrenginyje.

Programinės įrangos programavimo priemonių analizės tikslas yra surasti analizuotų įrankių skirtumus ir panašumus. Šio bandymo atlikėjas yra jaunesnysis Java programuotojas su pažengusiomis Java programavimo žiniomis. Vartotojas moka Java kalbą bei žino pagrindines Java bibliotekas. Taip pat turi patirties nedidelių projektų kūrimu su keletu paketu ir klasių.

Programinių paketų analizei pasirinktas pažintinio apėjimo metodas (angl. Cognitive walkthrough). Atliekant bandymus kiekvienas žingsnis yra įvertinamas užduodant keturis pagrindinius klausimus:

1. Ar vartotojas gali nuspėti kam reikalinga ši funkcija? Ar vartotojas supranta kokiam tikslui pasiekti yra naudojama ši funkcija?
2. Ar vartotojas pastebi, kad reikiamas veiksmas yra pasiekiamas?
3. Ar vartotojas supranta, kad reikiama funkcija gali būti įvykdoma su tam tikru mygtuku? Pvz. reikiamas mygtukas yra matomas, bet vartotojas nesupranta jo pavadinimo ir nespaudžia ant jo.
4. Ar vartotojas gauna atsakymą? Ar vartotojas sužino, kad jis atliko reikiamą užduotį panaudojus tam tikrą funkciją.

Programavimo įrankių analizė ir jų palyginimas yra labai specifinė užduotis, todėl kai kurios užduotys negali būti įvertinamos tik šiais keturiais klausimais. Nemažai funkcijų yra sukurtos neprikaištingai ir patikrintos laiko, todėl ne visada bus galima užduoti visus šiuos klausimus.

### **1.1. Java ME programavimo aplinka.**

„Java Platform Micro Edition“ kitaip žinoma, kaip Java ME (J2ME), yra Java platformos poaibis skirtas mobilių telefonų, delninių kompiuterių, išmaniųjų telefonų ir kitų mažų prietaisų programinės įrangos programavimui ir plėtojimui. Šis produktas buvo sukurtas kompanijos „Sun Microsystems“ ir jis yra „PersonalJava“ pakaitalas.

Java ME technologija iš pradžių buvo kuriama kartu su mažų prietaisų gamintojų asociacija. Dėl šios priežasties kompanija „Sun Microsystems“ apibrėžė technologijos Java Me pagrindus, tokius kaip ribotoje aplinkoje sukurti Java programą, kuri veiktų mažuose prietaisuose, turinčiuose ribotą atmintį, ekraną ir energijos talpą. [2]

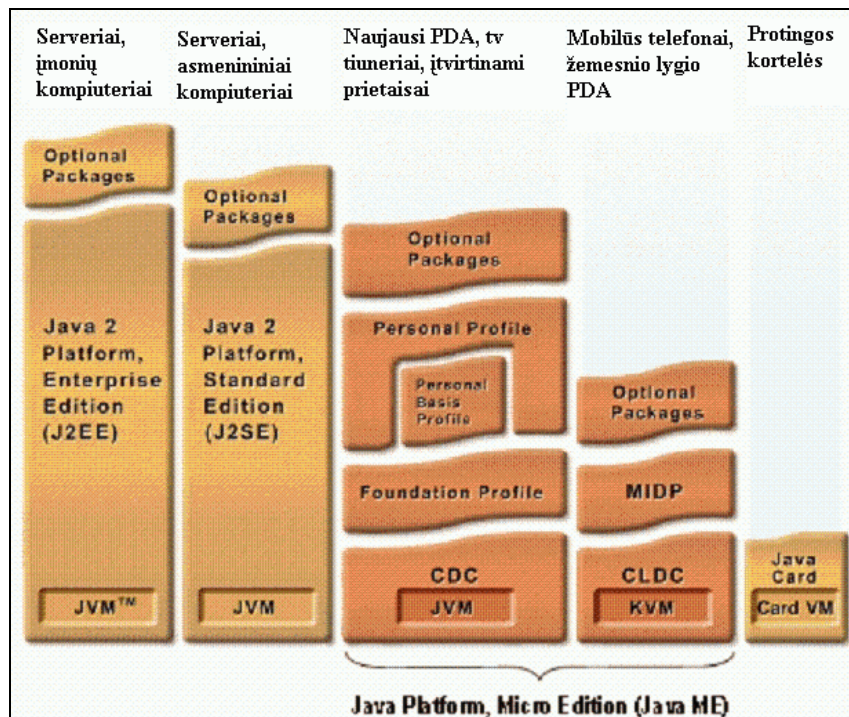
Ši platforma yra technologijų ir reikalavimų rinkinys, kurios gali būti sujungtos konstruoti užbaigtą Java aplinką, ypatingai atsižvelgiant į tam tikro prietaiso arba rinkos reikalavimus. Java ME suteikia lankstumo ir koegzistavimo visiems vartotojams. [2]

Java ME technologiją sudaro trys pagrindiniai elementai:

- Nustatymai suteikia didelį bibliotekų rinkinį ir virtualaus mechanizmo galimybes plačiai prietaisų įvairovei.
- Profilis yra sudarytas iš API rinkinio, kuris palaiko siauresnę prietaisų įvairovę.
- Pasirenkamas paketas yra technologinis – specifinis API rinkinys.

Šiuo metu Java ME yra perskirstyta į dvi pagrindines konfigūracijas, viena skirta paprastesnio techninio lygio mobiliems prietaisams, o kita skirta daugiau sugebantiems mobiliems prietaisams, tokiems kaip išmanieji telefonai arba delniniai kompiuteriai. Konfigūracija skirta paprastiems mobiliems prietaisams pavadinta „Connected Limited Device Configuration (CLDC)“, o gablesnė konfigūracija vadinasi „Connected Device Profile (CDC)“. [1]

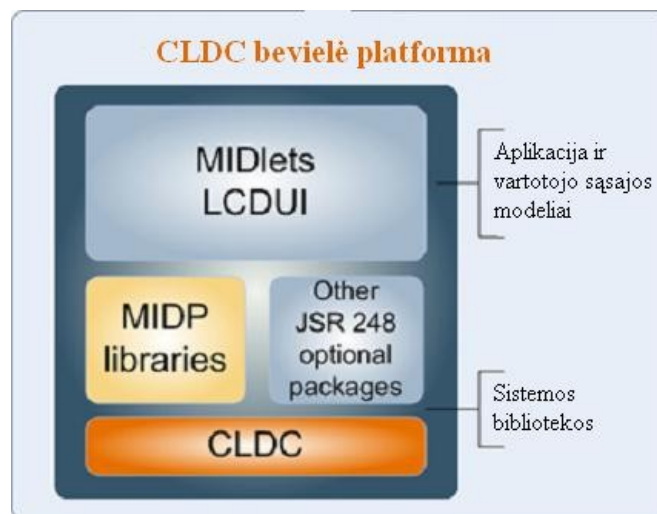
Žemiau pateiktame paveiksle matome komponentus iš kurių sudaryta Java platforma ir kurią dalį užima Java ME.



Šaltinis: <http://java.sun.com/javame/technology/index.jsp>

1 pav. Java platforma.

Java ME konfigūracija skirta mobiliems telefonams yra pavadinta „Connected Limited Device Configuration (CLDC)“. Ji specifiskai sukurta deranti su prietaisais veikiančiais Java platformoje, kurie turi ribotą atmintį, energijos šaltinį bei ribotas grafikos galimybes. Skirtingų Java ME konfigūracijų viršuje tiksliai nustatomas profilių skaičius apibrėžiantis aukštesnio lygio API, kuri toliau apibūdina taikomąsias programas. Plačiai priimtinas pavyzdys yra sujungti CLDC su „Mobile Information Device Profile (MIDP)“ tiekti užbaigtą Java taikomąją programą, mobilių telefonų ir kitų prietaisų su panašiomis galimybėmis, aplinkoje. [1]

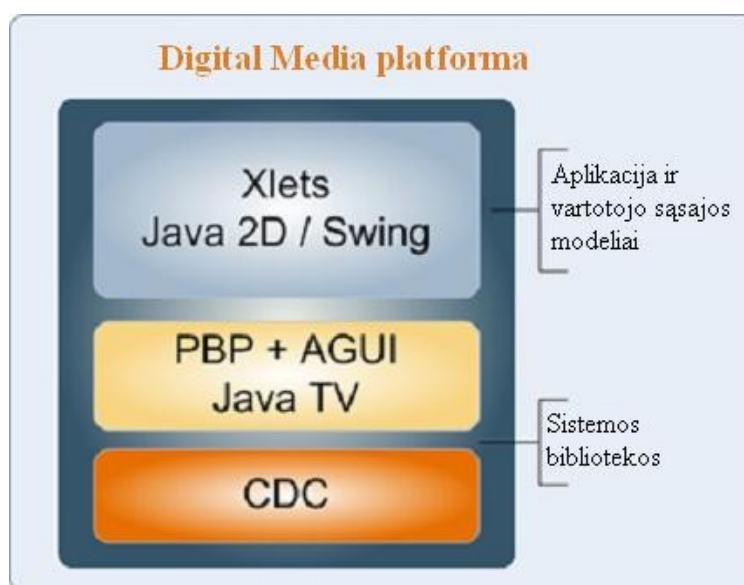


Šaltinis: <http://java.sun.com/javame/technology/index.jsp>

2 pav. CLDC platforma.

Java ME dėl CLDC ir MIDP aplinkos, kurios yra daugelyje mobilių prietaisų, yra puiki priemonė MIDlet kūrimui. MIDlet yra taikomosios programos sukurtos Java ME programinės įrangos plėtotojų, tokie kaip žaidimai, verslo programos bei kitos mobilios ypatybės. Šios programos viena karta parašytos gali būti paleidžiamos visuose prietaisuose, kurie palaiko Java ME technologiją. MIDlet gali būti patalpintas bet kurioje pasaulio saugykloje, o galutinis vartotojas ja paprasčiausiai gali parsisiusti ir naudoti savo prietaise. [1]

Konfigūracija, skirta didesniems, talpesniems bei sujungimo galimybę turintiems prietaisams, tokiems kaip išmanieji telefonai bei delniniai kompiuteriai yra pavadinta Connected Device Profile (CDC). Pagrindiniai CDC tikslai yra didinti technologijos galimybes ir plėtoti įrankius veikiančius ant Java SE (angl. Standard Edition) platformos.[1]



Šaltinis: <http://java.sun.com/javame/technology/index.jsp>

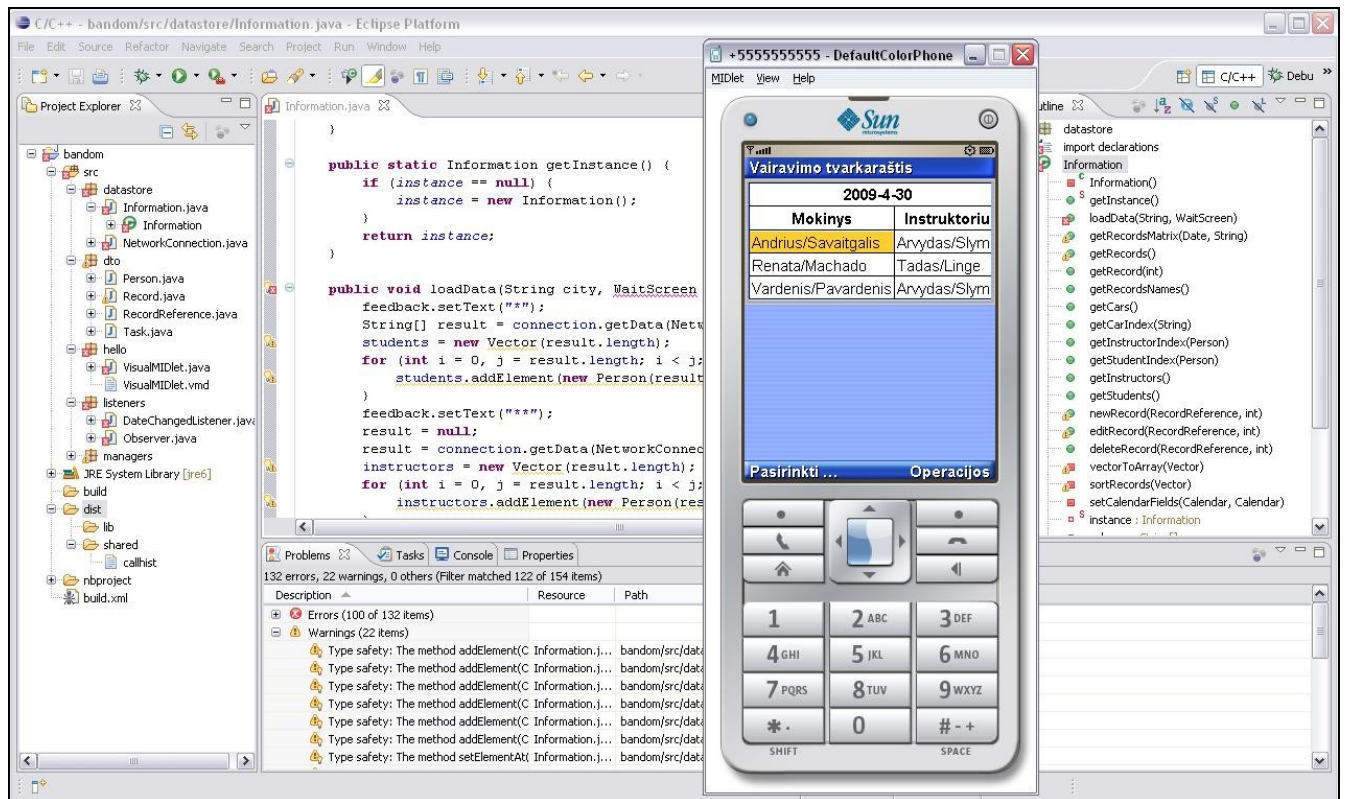
3 pav. Connected Device Profile (CDC)

CDC konfigūracijos teikiamos galimybės ir nauda:

- Įmonėms teikiama nauda naudojantis tinklo galimybėmis bei praplečiant verslą su mobiliais klientais, partneriais bei darbuotojais.
- Vartotojai gauna naudą dėl Java technologijos suderinamumo ir saugumo.
- Kūrėjams teikiama nauda iš Java programavimo kalbos saugumo, produktyvumo bei turtingos aplikacijų programavimo sąsajos (API) Java technologijoje.

## 1.2. Eclipse IDE

Eclipse programavimo aplinka (IDE) analizuojama dėl universalumo ir plataus pritaikymo, bei paplitimo tarp vartotojų. Tai yra kelių kalbų programinės įrangos platforma, kurią sudaro IDE bei išskiepiai skirti sistemos praplėtimui. Įrankis naudojamas Java aplikacijų kūrimui, be to įdiegus reikiamus iškiepius galima programuoti C/C++, Cobol, Python, Perl, PHP ir kitomis kalbomis. Šios platformos pagrindinis vaidmuo yra suteikti sklandžiai integruojamą priemonę vartotojams su jau naudojamais mechanizmais ir nustatytais taisyklėmis. [5]

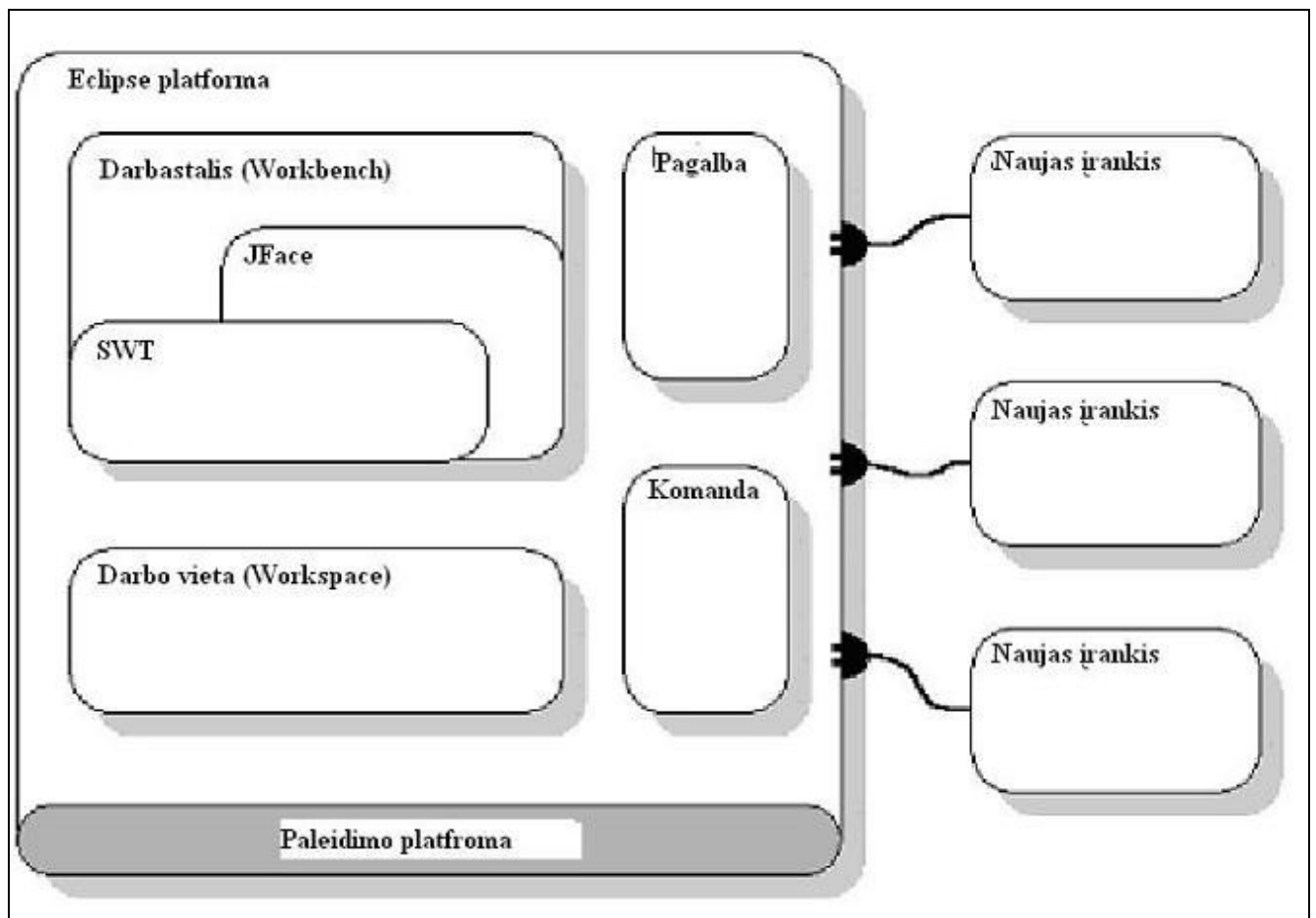


Šaltinis: Sudaryta autoriaus.

4 pav. Eclipse platformos vartotojo sąsaja.

Šis paveikslas vaizduoja pagrindinį darbatalio (angl. workbench) langą tik su standartiniais komponentais, kurie yra dalis Eclipse platformos. „Navigator“ vaizdas rodo failus vartotojo darbo vietoje (angl. workspace), teksto redaktorius rodo failo turinį, „Tasks“ vietoje vaizduojama reikalingų padaryti darbų sąrašas, „Outline“ vietoje vaizduojama redaguojamo failo ypatybės.

Eclipse platformos pagrindinis uždavinys yra vartotojams suteikti priemonę su mechanizmais ir taisyklėmis, kurie gali būti integruoti viename pakete. Šie mechanizmai yra pasiekiami per gerai apibrėžtą API sąsają, klases ir metodus. Platforma taip pat turi naudingus paruoštus blokus bei struktūras, kurie palengvins naujo įrankio plėtojimą. Žemiau esančiame paveiksle pavaizduoti pagrindiniai Eclipse platformos komponentai ir aplikacijų programavimo sąsajos (API). [5]



Šaltinis: <http://www.eclipse.org/whitepapers/eclipse-overview.pdf>

5 pav. Eclipse platformos architektūra.

Įskiepis (angl. Plug-in) yra mažiausia Eclipse platformos funkcijų dalis, kuri gali būti plėtojama ir pernešama atskirai. Paprastai nesudėtingas įrankis yra parašomas kaip vieno tipo įskiepis, taigi sudėtingo įrankio funkcijos gali būti paskirstomos keliems įskiepiams. Išskyrus nedidelį branduolio valdymą – paleidimo platformą, visas Eclipse platformos funkcionalumas yra įskiepiuose. Eclipse platformos sąsaja yra sukurta aplink darbastalį, kuris užtikrina bendrą struktūrą ir vartotojo sąsają.

Įskiepiei yra parašyti Java programavimo kalba. Tipiškas įskiepis susideda Java programinio kodo Java bibliotekoje, kelių „read-only“ bylų, paveikslukų, web šablonų, žinučių katalogų bei kitų bibliotekų. Taip pat yra mechanizmas, kuris leidžia sintezuoti įskiepi iš kelių fragmentų esančių tam tikrame kataloge arba interneto svetainėje. Kiekvienas įskiepis turi deklaracijos failą, skirtą nustatyti sujungimą su kitais įskiepiais. Įjungimo metu paleidimo platforma (angl. Platform Runtime) suranda turimus įskiepius, nuskaito jų deklaracijos failus ir sukuria įskiepių registrą atmintyje. Įskiepiei yra prieinami per API platformą, tačiau negali būti priedėti po programos paleidimo. Įskiepių deklaracijos failai saugomi specialiuose XML failuose, todėl deklaracijos informacija yra pasiekama iš įskiepių registro be jų aktyvavimo ar jų kodo užkrovimo. Tai yra svarbi savybė turint daug įdiegtų įskiepių.

tačiau vartotojo sesijos metu panaudojant tik kelis iš jų. Kol įskiepio kodas yra pakraunamas, jis turi nereikšmingą atminties žymą ir poveikio pradžios laiką. Įskiepis aktyvuojamas kai tikrai reikia paleisti jo kodą, t.y. kai vartotojas pasirenka pageidaujamą įskiepi iš sąrašo. Įskiepių aktyvavimas nevyksta automatiškai, tam naudojami keli specialūs API metodai.[5]

Eclipse platforma paleidžiama vienu kreipimosi į Java virtualiąją mašiną. Kiekvienas įskiepis yra priskirtas savo Java klasės kelties programai (angl. loader), kuri yra atsakinga už tų klasių įkėlimą. Kiekvienas įskiepis aiškiai deklaruoja savo priklausomybę kitiems įskiepiams ir taip jie tiesiogiai susiejami su klasėmis. Įskiepiei kontroliuoja viešų klasių matomumą bei sąveikas jų bibliotekose. Įskiepių mechanizmas yra reikalingas padalinti Eclipse platformą. Atskiri įskiepiei valdo darbo vietą (angl. workspace), darbatalį (angl. workbench) ir kitus. Netgi platformos paleidimas (angl. Platform Runtime) turi atskirą įskiepi.

Įvairūs įrankiai įdiegti į Eclipse platformą valdo įprastus failus vartotojo darbo vietoje (workspace). Ji sudaryta iš vieno arba daugiau top lygio projektų, kur kiekvienas projektas turi savo vietą vartotojo nurodytame kataloge. Projektų mechanizmas leidžia pažymėti suteikiant jam asmeniškumą arba tam tikrą bruožą. Kiekvienas projektas gali turėti tiek bruožų kiek reikia. Tai suteikia galimybę dalintis projektu neturint informacijos apie kiekvieną iš jų. Kiekvienas projektas sudarytas iš vartotojo sukurtų ir valdomų failų. Visi failai esantys darbo vietoje (angl. workspace) yra tiesiogiai prieinami standartinėms programoms ir įrankiais įdiegtais operacinėje sistemoje. Darbo vietos (angl. workspace) praeities mechanizmas padeda sumažinti netikėto failų praradimo riziką. Šis mechanizmas seka ir saugo failų turinio pasikeitimus. Pagal užimamos vietos ir datos nustatymus vartotojai kontroliuoja kaip yra tvarkoma praeitis. Darbo vieta (angl. workspace) turi žymėjimo mechanizmą, skirtą resursų žymėjimui. Žymėjimas yra naudojamas įvairių komentarų rašymui: kompiliavimo klaidos žinutės, numatomų padaryti darbų sąrašas, paieškos raktažodžiai bei derinimo programos žymos (angl. debugger breakpoints). Įskiepiei gali suteikti naujų žymeklių tipų bei kontroliuoti nesvarbu kur jie būtų saugomi. [5]

Eclipse platformos vartotojo sąsaja yra sukurta apie darbatalį (angl. workbench), kuris valdo visą struktūrą bei pristato vartotojui pratęstą vartotojo sąsają. Darbatalio (angl. workbench) API ir vykdymas yra sudarytas iš dviejų priemonių:

- SWT (Standard Widget Toolkit) pateikia bendrą nuo operacinės sistemos nepriklausančius valdiklius ir grafiką realizuojant tuo keliu, kuris leidžia glaudžią integraciją su pagrindine langų sistema. Visa Eclipse platformos vartotojo sąsaja ir priemonės prijungtos prie jos ir naudoja SWT perteikti informaciją vartotojams. Kiekvienai langų sistemai SWT įgyvendinimas. Jis naudoja turimus valdiklius, tačiau jų neturint SWT įgyvendinimas pasiima tinkamą emuliaciją.



Bendrieji žemo lygio valdikliai kaip sąrašas, teksto laukas arba mygtukai yra įgyvendinami bet kur, tačiau aukštesnio lygio valdikliai gali reikalauti langų pavyzdžių sistemoje. Ši strategija leidžia išlaikyti vienodą programavimo modelį visose aplinkose. SWT glaudžiai sąveikauja su operacinės sistemos darbastalio savybėmis „drag and drop“ bei gali naudotis operacinės sistemos komponentais.

- JFace yra vartotojo sąsajos priemonė su klasėmis, skirta tvarkyti daug bendrų sąsajos programavimo uždavinių. JFace yra nepriklausoma langų sistema ir yra skirtas darbui su SWT, jo nepaslepia. JFace apima įprastus sąsajos priemonių komponentus, tai – paveikslų ir šrifto registras, dialogas, nustatymai, vedlio pagrindas bei procesų ataskaitos.

SWT ir JFace yra bendros paskirties vartotojo sąsajos priemonės, tuo tarpu darbastalis (angl. workbench) suteikia Eclipse platformos asmeninę sąsają. Darbastalio API yra priklausoma nuo SWT API ir šiek tiek mažiau nuo JFace API (žr. 5 paveikslas). Eclipse platformos vartotojo sąsają sudaro redagavimas, peržiūra ir perspektyva. Iš vartotojo pozicijos darbastalio (angl. workbench) langas sudarytas iš peržiūros ir redagavimo. Perspektyva pasireiškia per redagavimo ir peržiūros pasirinkimą ir suderinimą. [5]

Redagavimas leidžia vartotojui atidaryti, redaguoti ir saugoti objektus. Atidarymo – saugojimo – uždarymo ciklas yra panašus kaip failų sistemose, bet yra glaudžiau integruotas į darbastalį (angl. workbench). Platforma suteikia standartinį teksto redaktorių, tačiau įdiegus papildomus įskiepius galima turėti specifiškesnį redaktorių.

Peržiūra teikia informaciją apie objektus su kuriais vartotojas dirba darbastalyje (angl. workbench). Peržiūra gali padėti redagavimo įrankiui teikiant informaciją apie tai koks dokumentas buvo redaguotas. Ji taip pat gali papildyti kitus peržiūros langus teikiant informaciją apie šiuo metu pažymėtus objektus. Pavyzdžiui, standartinė savybių peržiūra rodo objekto savybes pasirinktas kitoje peržiūroje. Peržiūros funkcija turi paprastesnį ciklą nei redagavimas: pakeitimai iškart išsaugomi ir atsispindi kitose vartotojo sąsajos vietose. Platforma turi keletą standartinių peržiūros funkcijų, o papildomos yra įdiegiamos su kitais įskiepiais. [5]

Darbastalio (angl. workbench) langas gali turėti kelias skirtingas perspektyvas, iš kurių tik viena yra matoma tam tikru momentu. Kiekviena perspektyva turi savo peržiūros ir redagavimo funkcijas, kurios išdėliojamos tam tikra tvarka, be to kai kurios iš jų gali būti paslėptos. Perspektyva kontroliuoja pradinę peržiūros funkcijų matomumą, išplanavimą ir veiksmų matomumą. Vartotojas gali paprastai perjungti perspektyvas darbui su skirtingomis užduotimis ir jas persitvarkyti pagal poreikius. Platforma turi standartines perspektyvas resursų tvarkymui, online pagalbą bei komandos palaikymo užduotis. Papildomos perspektyvos diegiamos kaip atskiri įskiepiai. [5]

Eclipse IDE turi Eclipse Mobile Tools for Java Project (MTJ) įskiepi, skirtą Java ME (Micro edition) aplikacijų kūrimui ir plėtojimui. Pagrindinės Eclipse IDE Mobile atliekamos funkcijos:

- J2ME MIDlet projektų kūrimas.
- J2ME MIDlet kūrimas.
- MIDlet paleidimas ir suderinimas.
- MIDlet pakavimas.
- Prietaisų valdymas.
- JAD ir JAR failų generavimas. [3]

Eclipse platforma turi branduolį bendrų blokų ir aplikacijų programavimo sąsają kaip darbo vietą (angl. workspace) ir darbatalį (angl. workbench) bei kitų praplėtimo punktų per kuriuos gali būti integruojamos papildomos funkcijos. Šie praplėtimo punktai, pagaminti kaip atskiri įskiepai, gali praplėsti Eclipse platformą ir suteiks galimybe kurti tam tikro tipo aplikacijas.

### 1.3. NetBeans IDE

NetBeans yra integruota programų kūrimo aplinka (IDE). Tai atviro kodo aplinka turinti visus reikiamus įrankius kurti profesionalias įprasto vartojimo ir įmonių programinę įrangą, web ir mobilias aplikacijas Java kalba ir C/C++, taip pat palaikomos dinaminės programavimo kalbos PHP, JavaScript, Groovy ir Ruby. NetBeans yra lengvai įdiegiama bei veikia daugelyje operacinių sistemų, Windows, Linux, Mac OS X, Solaris. Kiekvienas vartotojas gali atsisiųsti skirtingas programavimo įrankio konfigūracijas arba pasirinkti pilną paketo versiją. Atsisiuntus ne visus komponentus vėliau gali reikti juos įdiegti, o tai atliekama naudojantis NetBeans įskiepių langu. Šie įskiepai įdiegiami rankiniu būtu, tačiau programos atnaujinimai atsiumčiami automatiškai. [4]

NetBeans moduliai:

- **Base IDE** – Palaiko JavaScript ir CSS redagavimą, failų versijų kontrolė, unikali kūrėjų bendradarbiavimo funkcija.
- **Java SE** – šis modulis suteikia visus reikiamus įrankius, kurių reikia profesionaliai Java SE aplikacijai. NetBeans IDE palaiko plėtojamą su JDK 1.1, JDK 1.2, JDK 1.3, JDK 1.4, JDK 5.0 ir JDK 6.
- **WEB ir Java EE** – suteikia galimybę kurti įprasto naudojimo ir verslo programinę įrangą, kuri gali būti susieta su web naršykle per internetą arba intranetą.
- **Mobility (Java ME)** – leidžia kurti, testuoti, suderinti bei išskleisti programinę įrangą kuri veikia mobiliuose telefonuose, delniniuose kompiuteriuose arba TV prieduose, kuri yra suderinama su daugeliu prietaisų.
- **Ruby** – įrankis skirtas programuoti Ruby programavimo kalba bei vystoma su Ruby on Rails 2.0 programavimo terpe.
- **C/C++** - programavimo kalbos palaikymas, kuria galima kurti profesionalias pastovias aplikacijas Windows, Linux, Mac OS X, Solaris platformoms.
- **UML** – naudojant NetBeans UML modeliavima projektuotojai gali koncentruotis ties programinės įrangos projektavimu, o kūrėjai – ties programinio kodo rašymu.
- **SOA** – struktūrų orientuotų į paslaugą (SOA) aplikacijų rašymas, testavimas ir suderinimas naudojant XML, BPEL ir Java web servisus.

NetBeans programinį paketą plėtojanti kompanija „Sun“ ilgą laiką rinko atsiliepiamus apie senesnių versijų redagavimo įrankį. Surinkta informacija buvo pakankamai naudinga, todėl NetBeans IDE 6.5.1 turi pilnai išbaigtą redagavimo įrankio struktūrą. Jis turi patobulintą paryškinimų sistemą, t.y. specialus statinių kintamųjų žymėjimas, prieštaringi kodo elementai perbraukiami linija, paryškinti

klasių pavadinimai, kabučių, skliaustų užbaigimas ir kiti elementai. Šis redagavimo įrankio funkcionalumas yra galimas visose programavimo kalbose, kurias palaiko NetBeans IDE. Pastačius kursorių tarp skliaustų jie abu yra paryškunami, be to kursorių pastačius ant kintamojo jis paryškintas kita spalva. Dar viena svarbi savybė yra kintamojo vardo keitimas. Esant tokiam poreikiui naujas pavadinimas bus atnaujintas visame projekte. NetBeans įrankis turi paprastą kodo įterpimo sistemą. Rašant kodą yra generuojami visi to kodo elementai, parašius pradžių sistema automatiškai išdėsto kitus elementus (pvz. try...catch blokas). Be to sistema leidžia kodo elementų pasirinkimą iš iššokančio meniu. NetBeans suteikia galimybę atskirai matyti redagavimo langus bei laisvai valdyti meniu juostas. Ši funkcija yra labai tinkama kai vartotojas dirba su keliais monitoriais.

Svarbu kuo daugiau žinoti kaip kodas veikia, jo kūrimo metu. Netbeans turi derinimo programą (angl. debugger), kuri aptinka programuojamo kodo klaidas, išbando jį retose sąlygose bei vykdo nuolatinį testavimą. Programavimo įrankis turi matuoklę kiek laiko užima tam tikro klasės ar metodo įvykdymas bei kiek jos užima įrenginio atminties. Įjungus aplikaciją šis įrankis išbando aplikaciją ir po to pateikia grafiškai kuri klasė ar metodas truko ilgiausiai, kuris buvo dažniausiai iškviečiamas ir panašiai. Tai įrankis su kuriuo programuotojai turi būti labiausiai susipažinę.

NetBeans projektai dažnai reikalauja skirtingų sudėtinės konfigūracijos nustatymų. Kiekviena konfigūracija yra paprastai pasirenkama iš iškrentančio sąrašo, kuris yra pagrindinėje įrankių juostoje. Šiuo įrankiu galima kurti sudėtinės paleidimo konfigūracijas, kurios skirtingai veikia aplikaciją, naudoja skirtingus parametrus ir panašiai.

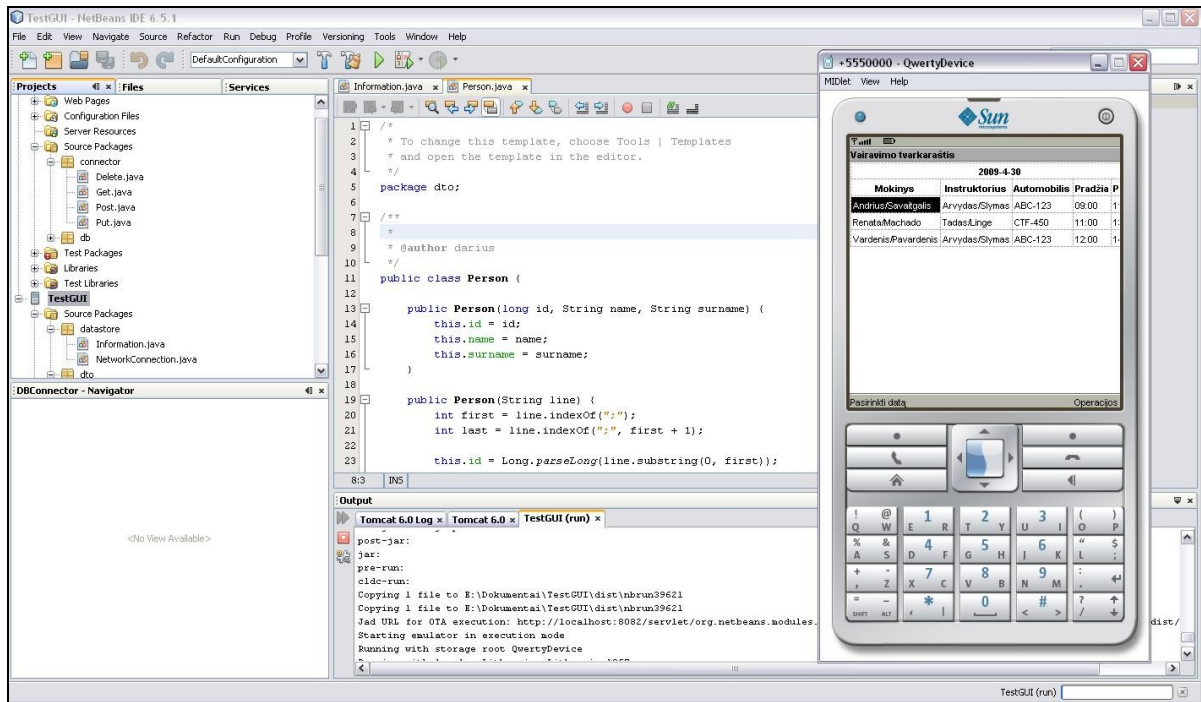
Netbeans funkcija „Local history“ saugo visus pakeitimus, kurie buvo padaryti tam tikrame projekto faile. Kai yra padaromas pakeitimas ši funkcija užfiksuoja ir išsaugo padarytus pakeitimus pagal laiką ir datą. Turint šią funkciją nereikia rizikuoti, kad bus prarastas kodas ar jo pakeitimai.

Programavimo įrankis turi laiką taupančius projektų vedlius, kurie sukuria infrastruktūrą tuzinams skirtingo tipo projektų. Projektų vedliai priskiria bibliotekas ir struktūrą, kurio projektas reikalauja ir dažnai generuoja keletą pradinių klasių. Naudojantis vedliu yra paprasta teisingai pradėti su visais reikalingais įrankiais ir bibliotekomis, kurios bus reikalingos kuriamai aplikacijai.

### **NetBeans Mobility**

NetBeans Mobility yra nepriklausomas NetBeans IDE platformos įrankis. Jis gali būti naudojamas aplikacijų, veikiančių plataus naudojimo mobiliuose prietaisuose su Java Micro Edition (Java ME) platforma, rašymui, testavimui ir suderinimui. Šis įrankis palaiko dvi pagrindines Java ME platformos konfigūracijas:

- Connected Limited Device Configuration (CLDC) yra konfigūracija skirta prietaisams su mažesne atminties talpa ir lėtesne duomenų apdorojimo sparta.
- Connected Device Configuration (CDC) – konfigūracija skirta prietaisams su daug didesne atminties talpa, didesne duomenų apdorojimo sparta ir tinklo galimybe. Tai išmanieji telefonai, TV priedai, komunikatoriai bei įmontuojami serveriai. [1]



Šaltinis: Sudaryta autoriaus.

6 pav. NetBeans Mobility (JavaME) programos langas.

- NetBeans Mobility įrankio savybės [4]:
- Projekto palaikymas, Apache Ant. Sutvarko MIDP programinį kodą ir projekto meta duomenis patogesniai valdymui. Taip pat leidžia kurti projektą už IDE ribų.
  - Integracija su J2ME Wireless Toolkit 2.2.
  - Prietaiso fragmentavimas – aplikacijos kūrimas sudėtiniais prietaisams ir prietaiso specifinio kodo vykdymas kaip konfigūraciją vienos aplikacijos viduje.
  - MIDP lokalizacijų palaikymas – leidžia pridėti specifinius komponentus į kuriamą aplikaciją.
  - J2ME MIDP 2.0 ir CLDC 1.1 standartų palaikymas
  - Emuliatorių platformų palaikymas – aplikacijos testavimas įvairių prietaisų emuliatoriuose, kuriuos teikia gamintojai.
  - Integruotas optimizavimo palaikymas – pridedamas saugumas, pagerinamas veikimas ir aplikacijos dydžio optimizavimas
  - Over-The-Air siuntimo testavimas – Emuliatorius simuliuoja tikro mobiliojo įrenginio veikimą.

## 1.4. Python for S60

Python yra interaktyvi programavimo kalba. Jos viena iš atmainų yra skirta išmaniųjų telefonų platformai Series 60 su Symbian operacine sistema mobiliems įrenginiams. Šis įrankis leidžia greitai plėtoti taikomąją programą arba jos prototipą. Taip pat jis leidžia sukurti visiškai nepriklausomą aplikaciją veikiančią S60 platformoje. Python kuriama kaip atviro kodo projektas. Python programavimo kalba leidžia naudoti keletą programavimo stilių: objektinį, struktūrinį, funkcinį, aspektinį. Python naudoja dinaminį tipų tikrinimą. Python kūrėjų tikslai buvo sukurti kalbą, kuri yra lengvai skaitoma, išraiškinga, išreikštinė, paprasta (tinkama neprofesionaliems programuotojams). Nors pradžioje ji buvo kuriama kaip scenarijų kalba, dabar ji naudojama ir dideliems programiniams projektams. Dabartinis Python for S60 yra pagrįstas Python 2.2.2 versija. Ji palaiko daugelį Python standartinės bibliotekos modulių, bet taip pat turi keletą mobilios platformos specifinių modulių [3]:

- Native GUI widgets,
- Bluetooth,
- GPRS networking,
- GSM vietos nustatymas,
- SMS žinutės,
- Kameros panaudojimas.

Programinė įranga veikia daugumoje operacinių sistemų: Windows, Mac OS X, Linux, Amiga, Palm OS, ji taip pat veikia Nokia S60 2nd ir 3rd versijos mobiliuose telefonuose, be to šia programavimo kalba sukurtos taikomosios programos veikia visuose Nokia Series 60 telefonuose (pvz. N-Gage QD, 6600, 6630, 6670, 7610, N70 ir N90). [3]



Šaltinis: Jurgen Scheible 2007 Mobile Python. 30p.

(a)

(b)

(c)

7 pav. Python kodo pavyzdžiai.

6a paveiksle vaizduojama Python programinis kodas, kuris yra toks pat kompiuteryje ir telefone, nors kompiuteris neturi priėjimo prie telefono funkcijų. 6b paveiksle matome Python for S60 veikianti telefone ir turinti priėjimą prie telefono funkcijų. 6c vaizduojama programa-interpretatorius, kuri naudojama kompiuteryje.

Python for S60 gali būti skirstoma į dvi plačias kategorijas:

1. Standartinės bibliotekos
2. Dinamiškai įkeliamos bibliotekos

Standartinės bibliotekos taip pat skirstomos į dvi modulius: e32 ir appuifw. E32 modulis suteikia prieigą prie servisu kurie yra neprieinami standartinėse Python bibliotekose. Iš esmės, šis modulis yra įdiegtas Python interpretatoriuje, kad suteiktų prieigą prie S60 platformos. Naudojantis appuifw moduliu vartotojas gali prieiti API susijusių su S60 platformos varotojo sąsaja. Pagrindiniai vartotojo sąsajos elementai yra teksto laukai (angl. Text), sąrašai (angl. Listbox), paveikslai (angl. Canvas) ir pan. Naudojantis šiais elementais GUI (grafinė vartotojo sąsaja) gali būti sudaroma kaip lango arba ekrano dalis. [3]

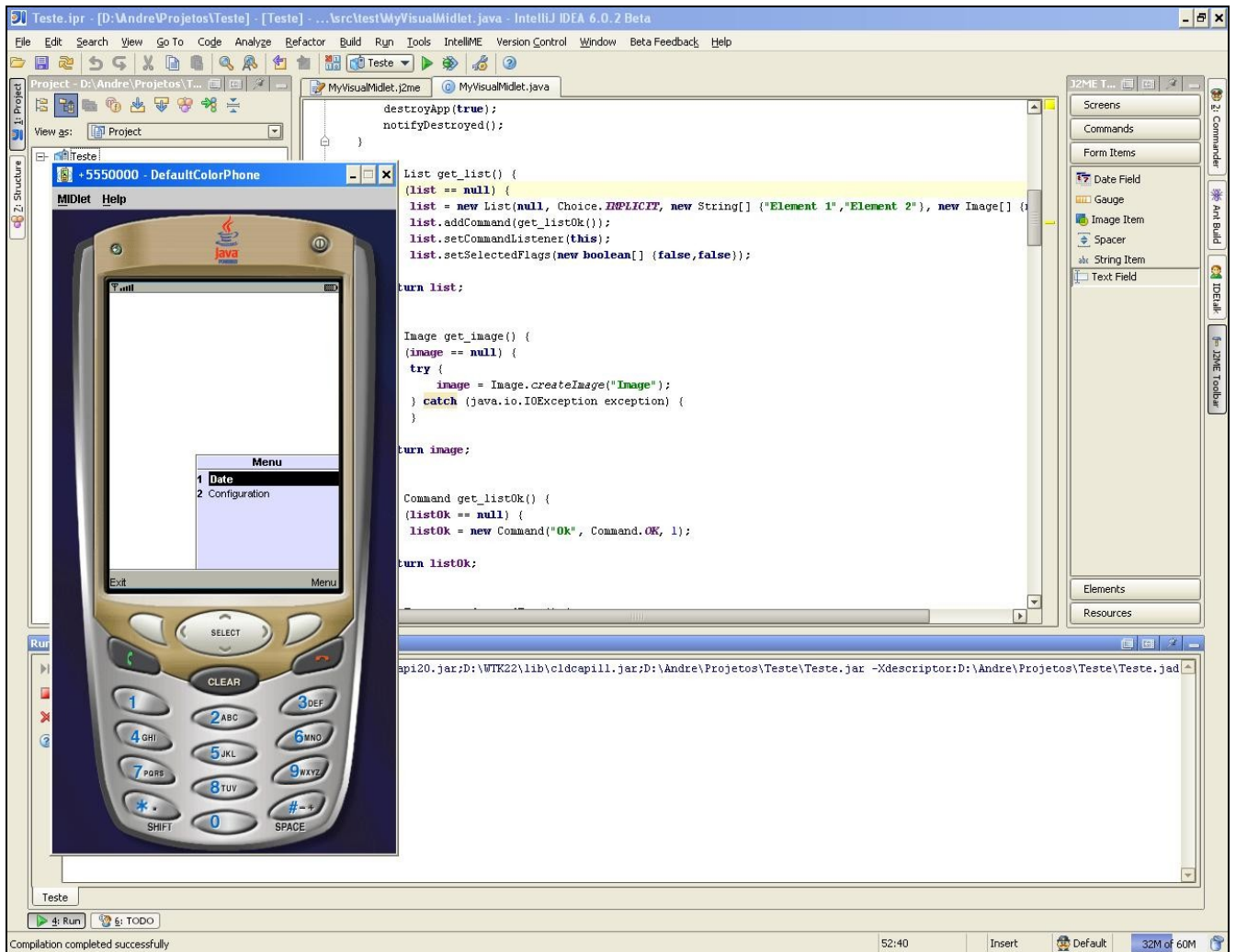
Dinamiškai įkeliamos bibliotekos yra užkraunamos tada kai reikia, priešingai standartinėms bibliotekom. To priežastis yra, kad dinamiškai įkeliamos bibliotekos turi priėjimą prie privačių S60 API. Viso yra 16 dinamiškai įkeliamų bibliotekų, tačiau dažniausiai naudojamos šios:

- Grafika – suteikia S60 platformos grafinių vaizdų valdymo galimybes.
- Žinutės – SMS ir MMS paslaugos.
- Kameros modulis – suteikia priėjimą prie visų kameros atliekamų funkcijų.
- Garsas – skirtingų formatų audio failų valdymas.
- Kalendorius – leidžia naudoti visas įrenginio kalendoriaus funkcijas.
- Kontaktų modulis – suteikia priėjimą prie įrenginyje saugomų kontaktų sąrašo.

Python Installer įrankis leidžia vartotojams į telefonus įdiegti Python skriptus ir bibliotekas. Šis įrankis yra įdiegiamas kartu su programavimo įrankiu. Jis priima .py, .pyc, pyo ir .pyd tipo failus

## 1.5. IntelliJ IDEA

IntelliJ IDEA yra Java integruota programų kūrimo aplinka (IDE), kuri turi labai patogius įrankius programavimo konstrukcijų automatizavimui, defektų paieškai bei kodo struktūros keitimo (angl. refactoring) funkcijas. Programinio kodo struktūros keitimo funkcijos leidžia lengvai keisti suprojektuotas programinio kodo paketų ir klasių struktūras, todėl nebūtina detaliai projektuoti prieš pradant programavimą.[6]



Šaltinis: Sudaryta autoriaus.

8 pav. IntelliJ IDEA programos langas.

IntelliJ IDEA turi galingą Java projektų ir failų valdymo aplinką. Ji palaiko populiariausius standartus: J2EE, J2ME, Ant, JUnit, AJAX ir kt., be to šis paketas palaiko daugelį programavimo kalbų: Java, JavaScript, HTML, XML, Ruby, Groovy, SQL ir kt. Programavimo paketo vartotojo sąsają sudaro trys pagrindiniai elementai: redagavimo langas (vieta kurioje matome kodą), projekto langas (šioje vietoje matome projektą sudarančius elementus) ir struktūros langas (lange matome atidaryto objekto informaciją).



Darbas su IntelliJ IDEA prasideda su projekto idėja. Projektą apima visi programinio kodo ir bibliotekų failai bei kūrimo instrukcijos viename organizaciniame vienete. Projekto lange matome šiuo metu atidarytus projektus. Projektai neturi programinio kodo, skriptų ar dokumentacijos, jie tik apibūdina nustatymus per kuriuos IDE kreipiasi į modulius ar bibliotekas. Šie nustatymai saugomi XML tipo failuose, jie paprastai valdomi, redaguojami be to lengva keistis konfigūracija su kitais projektais. Programavimo įrankis sukuria projekto, darbo vietos (angl. workspace) ir modulio failus. Projekto failai turi .IPR galūnę ir saugo pagrindinę projekto informaciją: sudarančių modulių pavadinimai ir adresai, kompiliatoriaus nustatymai, Ant nustatymai ir kt. Šis failas įjungia projektą ir pagal nutylėjimą yra sukuriamas pagrindiniame projekto kataloge. Šalia kiekvieno projekto failo yra sukuriamas .IWS failas, kuriame saugomi asmeniniai darbo vietos nustatymai. Šis failas atsimena langų sudėliojimą ir pozicijas, VCS ir istorijos nustatymus, Run/Debug konfigūracijos objektus bei kitą informaciją susijusią su programavimo aplinkos plėtojimu. Modulių failai sukuriami kiekvienam nustatytam moduliui ir jie turi .IML galūnę. Jie saugo visą informaciją susijusią su tam tikru moduliu, o jo turinys priklauso nuo konkretaus modulio tipo. [6]

IntelliJ IDEA turi puikų programinio kodo redaktorių. Automatinis kodo pabaigimas veikia greitai ir užbaigia netgi tuos kodo elementus kurių užbaigimas nesitikimas, be to jis suranda ir praneša apie kodo klaidas. Kodo redaktorius turi protingą pagalbą, kuri seka programuotojo veiksmus ir tam tikrose situacijose pateikia pasiūlymus tolesniems veiksams. Kai rašomas programinis kodas IntelliJ IDEA visada yra vienu žingsniu priekyje, analizuoja kodą ir greitai ištaiso daugelį klaidų arba duoda tam tikrus nurodymus. Kaip ir kiti IDE programavimo įrankiai IntelliJ IDEA turi puikias sintaksės paryškinimo funkciją. Be to jis leidžia pasirinkti programuotojui tinkančias spalvas. Jis leidžia paprastai generuoti naujas klases, sąsajas, paketus bei standartinius metodus. Taip pat IntelliJ IDEA išskirtinai leidžia dirbti klaviatūra nešvaistant laiko kol pele bus pasiektas tam tikras mygtukas.

Kaip minėta prieš tai IntelliJ IDEA geba atlikti kodo struktūros keitimą (angl. refactoring). Pavyzdžiui galima greitai perkelti klasę tarp paketų arba išimti tam tikrą klasės sąsają. Taip pat galima pakeisti klasės pavadinimą ir jis pasikeis visose klasėse kurios naudoja pakeistos klasės pavadinimą. Skirtingai nei kituose IDE įrankiuose IntelliJ IDEA keitimas vykdomas iššokančiame lange ir tai užtikrina sėkmingą kodo pakeitimą.

Šis IDE produktas siūlo plataus bendradarbiavimo komandinio darbo aplinką, kuri turi glaudžią integraciją su populiariausiomis versijų kontroliavimo sistemomis (CVS, Perforce, Subversion, StarTeam, Visual SourceSafe ir ClearCase). Be to pakete yra TeamCity funkcija, kuri programuotojų komandoms padeda sujungti pakeistas programinio kodo dalis. Versijų kontroliavimo sistemų

integracija su TeamCity leidžia programuotojams keistis tekstinėmis žinutėmis, praktiniais patarimais, programinio kodo patarimais bei pačiu kodu nepaliekant IDE aplinkos.

Lankstus konsolės išdėstymas, kintamųjų, sandaros ir sekimo dalys padaro derinimo funkcijos (angl. debugger) langą vartotojiškesnį bei supaprastina derinimo procesą. Sumanus derinamo kodo pasirinkimas leidžia pasirinkti bet kurią kodo dalį. Individualiai pasirenkamos objektų žymės leidžia susikoncentruoti į derinimo procesą ir atskirai sekti skirtingus objektus.

IntelliJ IDEA įdiegė projekto klasių priklausomumo analizės įrankį - Dependency Structure Matrix (DSM). Jis buvo sukurtas analizuoti sudėtinių projektų struktūrą su daugybe Java klasių. DSM vaizduoja tikrą projekto klasių tarpusavio priklausomumą. Šiuo įrankiu greitai galima surasti bet kurią projekto vietą, surasti priklausomumo ciklą, nenaudojamas klases ir pan.

IntelliJ IDEA turi integruotą Java ME Doja/MIDP/CLDC aplikacijų kūrimo įrankį. Sumanus programavimo padėjėjas atpažįsta aibę įvairių mobilių įrenginių SDK emuliatorių. Ši IDE įrankis palaiko visas tradicinio Java kodo redaktoriaus funkcijas. Java ME modulis kuriamai aplikacijai leidžia patogiu būdu nustatyti reikalingą SDK, aplikacijos resursus bei kitus nustatymus.

Skirtingai nei kiti analizuoti mobilių įrenginių programavimo įrankiai šis yra mokamas. Iš gamintojo (JetBrains) tinklapiio galima parsisiųsti 30 dienų bandomąją versiją, kuri tinkama Windows, Mac OS X bei Linux operacinėms sistemoms (<http://www.jetbrains.com/idea/download/index.html>).

**1 lentelė. IntelliJ IDEA licenzijų kainos**

<b>Licenzijos tipas</b>	<b>Naujo vartotojo kaina</b>	<b>Atnaujinimo kaina</b>
Komercinė licenzija. Kompanijoms ir organizacijoms	€528	€264
Asmeninio naudojimo licenzija. Pavieniai programuotojai.	€220	€132
Akademinė licenzija. Studentams ir dėstytojams	€88	€52
Klasės licenzija. Mokymo institucijoms	Nemokama	Nemokama
Atviro kodo licenzija. Atviro kodo projektams	Nemokama	Nemokama

Saltinis: <http://www.jetbrains.com/idea/buy/index.jsp>

Atnaujinimo licenzija reikalinga tik vartotojams, kurie turi 7 arba žemesnę IntelliJ IDEA versiją. Nemokamą licenziją turintys vartotojai ta pačia licenzija gali naudotis ir aštuntoje produkto versijoje.

## 1.6. Programinių įrankių palyginimas

Ištyrus Eclipse IDE, NetBeans IDE, Python for S60 ir IntelliJ IDEA programavimo priemonių galimybes, jų trūkumus ir privalumus jos bus sulyginotos tarpusavyje. Šie įrankiai lyginti palyginimo metodu. Palyginimo kriterijai pasirinkti pagal mobilių įrenginių programuotojų naudojamą funkcijas.

**2 lentelė. Programavimo įrankių palyginimas.**

	<b>Eclipse IDE</b>	<b>NetBeans IDE</b>	<b>Python for S60</b>	<b>IntelliJ IDEA</b>
Kalbos palaikymas	C/C++, Cobol, Python, Perl, PHP, Java	C/C++, Java, PHP, JavaScript, Groovy, Ruby	Python, C/C++	Java, JavaScript, HTML, XML, Ruby, Groovy, SQL
Sintaksės/ semantikos paryškinimas	+	+	+	+
Klaidų paryškinimas	+	+	+	+
Raktažodžių pabaigimas	+	+	-	+
Kodo formatavimas	+	+	+	+
Skliaustų, kabučių „end“ pabaigimas	-	+	-	+
Gramatikos tikrinimas (kodinių žodžių taisymas)	+	+	-	+
Kodo sutraukimas	-	+	-	-
Projekto valdymas	+	+	-	+
Metodų peržiūra	+	+	-	+
Klasių projektavimas	+	+	+	+
Lokali istorija (pakeitimų failuose saugojimas)	+	+	-	+
Derinimo programa (Debugger)	+	+	-	+
Atminties užimtumo, vykdymo laiko matuoklė	-	+	-	+
Vedliai (Wizards)	+	+	-	+
Atskiro failo redagavimas	-	+	-	+
WEB aplikacijų palaikymas	+	+	+	+
Versijų kontroliavimo sistema	+	+	-	+
Komandinio darbo įrankiai	-	-	-	+
Virtualaus mobilaus įrenginio palaikymas	-	+	-	+
Veikia mobiliame įrenginyje	-	-	+	-
Kaina ir atviro kodo galimybės	Atviro kodo.	Atviro kodo.	Atviro kodo.	Mokama.

Šaltinis: sudaryta autoriaus.

NetBeans IDE, Eclipse IDE ir IntelliJ IDEA yra integruotos programų kūrimo aplinkos turinčios patogų dizainą, tačiau NetBeans yra intuityvesnė nei kiti analizuoti įrankiai. Netgi pirmą kartą kuriant projektus, įterpiančiomus papildomus resursus, kuriant ir taisant programinį kodą nereiks vartyti vadovėlio ar naudotis pagalba, tačiau atlikti tuos pačius veiksmus kitais įrankiais iškiltu tam tikrų kliūčių nors Eclipse IDE ir IntelliJ IDEA turi panašias funkcijas, o kai kuriais kriterijais ir pranašesnės.

Iš šių keturių programavimo priemonių savo paprastumu ir mobilumu ypatingai išsiskiria Python for S60. Šis įrankis neturi sudėtingų resursų matuoklės funkcijų ar įmantraus kodo redaktoriaus, tačiau šis įrankis gali veikti pačiame mobiliame įrenginyje. Programinio kodo sintaksė yra ypač supaprastinta ir nesunkiai suprantama eiliniam vartotojui.

Kuriant universalios sąsajos taikomąsias aplikacijas geriausia naudotis integruotomis programavimo priemonėmis, t.y. NetBeans IDE, Eclipse IDE bei IntelliJ IDEA. Šie įrankiai turi daugiafunkcinius programinio kodo redaktorius, aplikacijų optimizavimo įrankius bei specialias, mobilių aplikacijų kūrimui skirtas funkcijas. Šioje srityje NetBeans IDE ir IntelliJ IDEA yra pranašiausi, nes jie turi atminties užimtumo ir funkcijų vykdymo laiko matuokles bei turi integruotą virtualų mobilų įrenginį. Vienintelis ir kone didžiausias NetBeans IDE pranašumas yra tai, kad tai yra nemokamas programavimo įrankis, todėl šis įrankis pasirinktas atlikti universalios sąsajos aplikacijos kūrimo eksperimentą.

## 2. PROJEKTINĖ DALIS

Projektinė dalis apima kuriamos taikomosios aplikacijos pasirinkimo priežastis, funkcijų nustatymą, architektūros sudarymą bei patį universalios sąsajos taikomosios aplikacijos projektą. Projekto dalyje konkretizuojamos ir grafiškai pateikiamos kuriamos aplikacijos funkcijos ir architektūra. Tai leidžia susidaryti pilną būsimos aplikacijos vaizdą.

### 2.1. Projekto tikslas

Projekto tikslas – praktiškai pritaikyti vienos iš mobilių įrenginių programavimo priemonių galimybes bei sukurti universalios sąsajos veikiančią aplikaciją, kuri būtų naudojama įmonės veikloje. Šiam tikslui pasiekti bus susisteminta mobilių įrenginių programavimo priemonių galimybių analizės metu gauta informacija, grafiškai atvaizduotos funkcijos bei sumodeliuotos norimos kompiuterizuoti funkcijos.

Šiuo projektu siekiama perkelti į mobilių įrenginių vairavimo mokyklos informacinės sistemos klientų registravimo praktiniam vairavimui funkciją. Sėkmingas jo įgyvendinimas duos ne tik ekonominės naudos vairavimo mokyklai, bet ir mokslinės naudos kuriant universalios sąsajos taikomąją aplikaciją.

Taikomoji aplikacija suteiks galimybę vairavimo instruktoriams bei kitiems vairavimo mokyklos darbuotojams užregistruoti klientą praktiniam vairavimui naudojantis mobiliu įrenginiu. Kadangi klientų srautas yra didelis ir vairavimo mokykla veikia keturiuose miestuose, todėl tokia taikomoji aplikacija padeda išvengti tokių klaidų kada klientas užregistruojamas kelis kartus arba tam pačiam laikui užregistruojami keli klientai. Kadangi duomenys realiu laiku atnaujinami duomenų bazėje, vairavimo mokyklos vadovai gali patogiai įvertinti vairavimo instruktorių užimtumą.

Šiuo laikotarpiu realaus laiko duomenų turėjimas yra vienas iš svarbiausių sėkmingos įmonės egzistavimo kriterijų. Būtent dėl šios priežasties pasirinkau sukurti mobilios įrenginio taikomąją aplikaciją, kuri susijungtų su duomenų baze ir joje sukurtu naujus, ištrintu arba atnaujintu jau egzistuojančius įrašus. Programavimo priemonių analizės skyriuje išanalizavus mobilių įrenginių programavimo priemones taikomosios aplikacijos kūrimui pasirinkau NetBeans programavimo įrankį. Tarp šio įrankio galimybių yra web aplikacijų kūrimas. Tai ypač reikalinga programavimo įrankio galimybė, nes mobilus įrenginys tiesiogiai negali jungtis prie duomenų bazės.

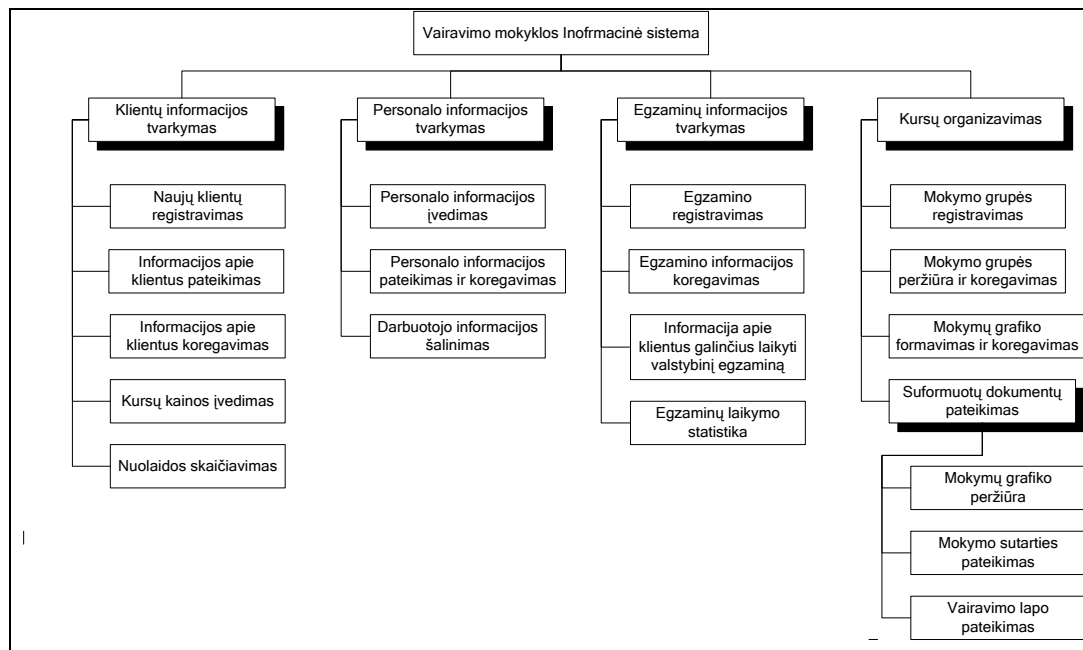
Rinkoje yra daug įvairių mobilių įrenginių. Skirtingi gamintojai naudoja skirtingas technologijas, todėl ta pati taikomoji aplikacija yra įvairiai atvaizduojama skirtinguose įrenginiuose. Taip pat gali būti tokių mobilių įrenginių kurie neatlieka visų taikomosios aplikacijos funkcijų arba jas vykdo tik iš dalies. Sukurtą taikomąją aplikaciją išbandysiu įvairiose mobilių įrenginių platformose.

Šiuo tyrimu sieksiu išsiaiškinti kuriuose mobiliuose įrenginiuose taikomoji aplikacija neveikia sklandžiai, kokios esančių sutrikimų priežastys bei sieksiu pašalinti surastas klaidas.

## 2.2. Kompiuterizuojamų funkcijų charakteristika

2007 metais buvo sukurta vairavimo mokyklos informacinė sistema. Vairavimo mokykla turi keturis padalinius, kuriuose yra įdiegta informacinė sistema. Informacinė sistema apima šias kompiuterizuojamas funkcijas:

- Klientų informacijos tvarkymas – pagrindinė funkcija. Ši funkcija apima klientų (mokinių) registravimą vairavimo kursams, jų informacijos saugojimą, apdorojimą bei atrinkimą pagal pasirinktus kriterijus.
- Personalo informacijos tvarkymas – papildanti funkcija. Ji apima teorijos mokytojų ir vairavimo instruktorių duomenų apdorojimą, saugojimą bei atrinkimą. Būtina norint realizuoti kursų organizavimo funkciją, nes reikia kontroliuoti kursų organizavimo tvarką bei teisingai paskirstyti
- Egzaminų informacijos tvarkymas – pagrindinė funkcija. Egzamino registravimas, jo informacijos saugojimas ir apdorojimas.
- Kursų organizavimas – tikslinė funkcija. Apima mokinių, grupių ir kursų duomenų tvarkymą, rezultatų apskaitą ir kursų rengimo sistemą. Kompiuterizuojama kaip pagrindinė organizacijos veiklos kryptis.



Šaltinis: Sudaryta autoriaus.

9 pav. Informacinės sistemos funkcijų hierarchijos diagrama.

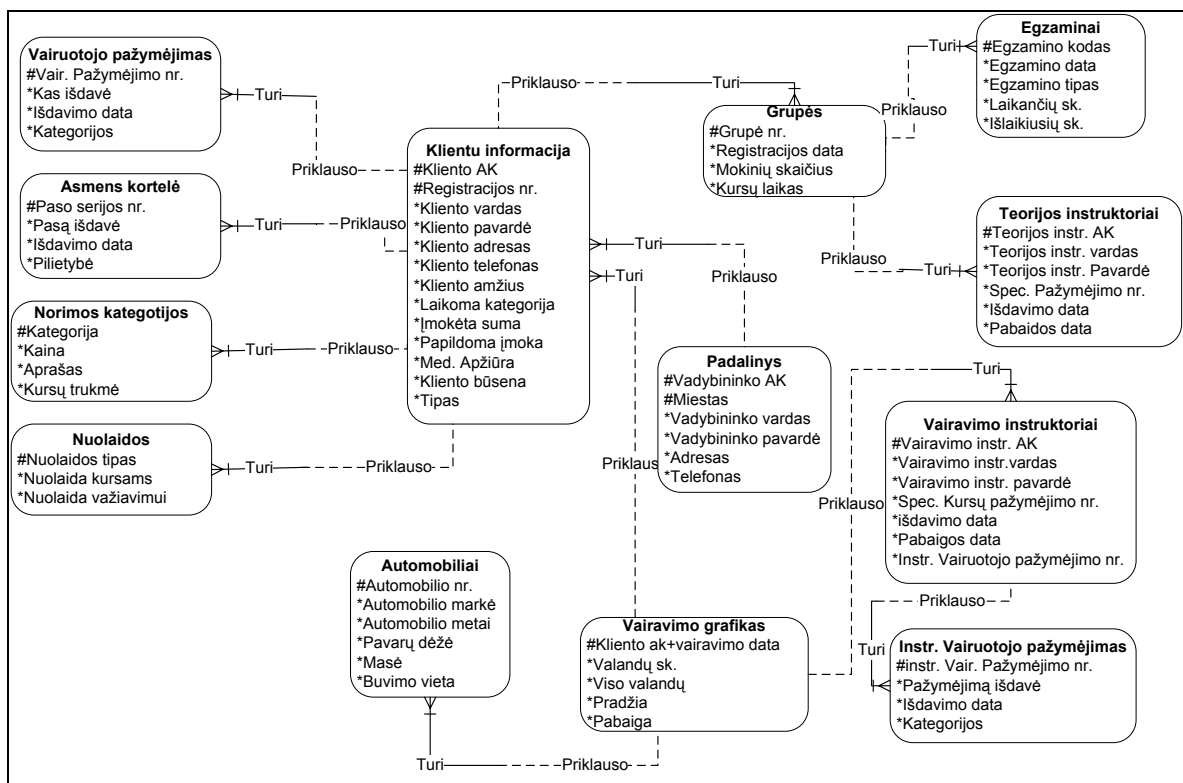
Siekiant modernizuoti vairavimo mokyklos veiklą, optimizuoti bei pagerinti darbuotojų darbą buvo sukurta mobili praktinių vairavimų tvarkymo programinė įranga. Mobiliam įrenginiui skirta programinė įranga turi šias funkcijas:

- Kliento registravimas praktiniam vairavimui.
- Užregistruoto praktinio vairavimo redagavimas.
- Užregistruoto praktinio vairavimo šalinimas.

Taikomoji aplikacija, įdiegta mobiliajame įrenginyje susijungs (naudojant GPRS Internetą) su vairavimo mokyklos duomenų baze ir joje talpina naujus duomenis bei atnaujins jau esančią informaciją. Ši programa suteiks galimybę vairavimo mokyklos darbuotojams net neturint šalia kompiuterio tvarkyti klientų praktinio vairavimo informaciją.

### 2.3. Duomenų bazės aprašymas

Vairavimo mokyklos informacinė sistema duomenis saugo MySQL duomenų bazėje. Konceptinis jos modelis pavaizduotas žemiau esančiame paveiksle.



Šaltinis: sudaryta autoriaus

10 pav. Konceptinis duomenų bazės modelis.

Programinė įranga atrinks duomenis iš „Kliento informacija“, „Automobiliai“, „Vairavimo instruktoriai“ ir „Vairavimo grafikas“ objektų. Duomenys bus saugomi ir atnaujinami „Vairavimo grafikas“ objekte.

Šiame modelyje vaizduojami objektai, jų atributai taip pat nurodomos kai kurios objektų savybės bei ryšiai tarp jų. ER diagramoje identifikuojami tokie objektai:

- Klientų informacija
- Vairuotojo pažymėjimas
- Asmens kortelė
- Norimo kategorijos
- Nuolaidos
- Egzaminai
- Grupės
- Teorijos mokytojai
- Padaliniai
- Vairavimo instruktoriai
- Vairavimo instruktoriaus vairuotojo pažymėjimas
- Automobiliai
- Vairavimo grafikas

#### **2.4. Priemonės naudotos programinės įrangos kūrimui:**

Mobilus įrankis buvo kuriamas naudojantis NetBeans IDE 6.5 programavimo paketu. Šis paketas pasirinktas prieš tai išanalizavus NetBeans IDE, Eclipse IDE, Python for S60 bei IntelliJ IDEA programavimo paketų galimybes ir funkcijas. Be šios integruotos programavimo aplinkos buvo naudojama papildoma programinė įranga:

- Microsoft Windows XP operacinė sistema.
- J2ME Wireless Toolkit 2.2 – Mobilioji Java
- WAMP5 Server programinis paketas.
- Apache Tomcat 6.0 serveris.

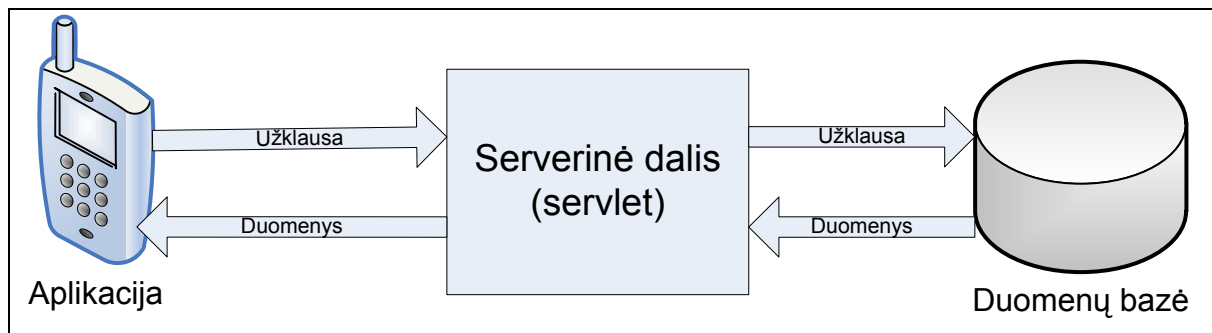


## 2.5. Programinės įrangos aprašymas

Programinė įranga yra sudaryta iš dviejų dalių – serverinės dalies (servlet) ir aplikacijos. Servletas patalpintas įmonės serveryje, kartu su duomenų baze, o taikomoji aplikacija bus diegiama vartotojų (vairavimo instruktorių) mobiliuose įrenginiuose.

### Serverinė dalis

Java Servlet – Sun Microsystems sukurta technologija, skirta dinaminiam turiniui generuoti. Tai serveryje esantis komponentas, gaunantis iš kliento (pvz., naršyklės, mobiliojo įrenginio) užklausa (angl. request) bei generuojantis atsaką (angl. response). Servletas gali būti generuojamas automatiškai, pasitelkiant JavaServer Pages technologiją ar kitą šablonų generavimo variklį [8].



Šaltinis: sudaryta autoriaus

11 pav. Programinės įrangos struktūra.

Mobilus įrenginys negali tiesiogiai jungtis prie duomenų bazės. Ši problema sprendžiama sukūrus servletą. Tai serveryje esantis komponentas, kuris gauna užklausas iš mobilus įrenginio, jungiasi prie duomenų bazės ir pagal gautą užklausa atrenka duomenis. Gauti duomenys yra siunčiami atgal į mobilų įrenginį ir atvaizduojami vartotojui. Servletą sudaro keturi metodai:

- Delete – trina pasirinktą įrašą.
- Get – pareikalauja informaciją iš serverio.
- Post – siunčia naujus duomenis į serverį.
- Put - siunčia atnaujinamus duomenis į serverį.

Pagal aplikacijos užklausas šie metodai sukuria bei tvarko jau esamus duomenų bazės įrašus.

### Taikomoji aplikacija

Aplikacija įdiegta varotojo mobiliame įrenginyje veiks sklandžiai jeigu serveryje bus duomenų bazė, servletas bei Tomcat serveris. Mobilūs įrenginiai turi skirtingo dydžio ekranus bei naudoja skirtingas technologijas, todėl skirtinguose įrenginiuose aplikacijos vaizdas skirsis bei gali neveikti keletas atliekamų funkcijų. Šie sutrikimai bus šalinami atliekant papildomus tyrimus.

Įjungus taikomąją aplikaciją būtina teisingai suvesti varotojo vardą ir slaptažodį. Tai atlikus aplikacija pareikalauja pasirinkti miestą ir datą, pagal kuriuos bus ieškomi užregistruoti praktiniai

vairavimai arba bus įvedamas naujas praktinis vairavimas. Tai atlikus, mobiliojo įrenginio ekrane pamatysim tvarkaraščio formą (duomenys pagal prieš tai pasirinktus parametrus). Įvykus duomenų perdavimo sutrikimams, vartotojas gauna klaidos pranešimą. Turint vairavimo tvarkaraštį galima atlikti naujo įrašo sukūrimą, esamo įrašo redagavimą arba šalinimą. Kuriant naują įrašą visų pirma patikrinama ar gali būti sukurtas naujas įrašas (t.y. ar data atitinka keliamus kriterijus). Jei patikrinimas sėkmingas vartotojui parodoma naujo įrašo įvedimo forma, priešingu atveju vartotojas gauna pranešimą ir iškviečiama tvarkaraščio forma. Vartotojui suvedus naujus duomenis ir pasirinkus juos saugoti, jie siunčiami į servletą ir jis išsaugo juos duomenų bazėje, o vartotojui atidaroma tvarkaraščio forma. Nepavykus išsaugoti duomenų vartotojas gauna pranešimą apie įvykusią klaidą. Pasirinkus redaguoti duomenų bazėje egzistuojantį įrašą, visų pirma patikrinama ar įrašo data nėra vėlesnė už redagavimo datą. Jei patikrinimas sėkmingas atidaroma įrašo redagavimo forma, priešingu atveju vartotojas pamatys pranešimą, kad įrašo redaguoti negalima ir bus grąžintas į tvarkaraščio formą. Atlikus reikiamus koregavimus duomenys išsaugomi duomenų bazėje taip pat kaip saugant naują įrašą. Šalinant pasirinktą įrašą tikrinama ar jis nėra vėlesnis už įrašo šalinimo datą, jei sąlygos atitinka įrašas yra pašalinamas iš duomenų bazės, naudojant servletą. Tačiau jeigu tikrinimo sąlygos neatitinka vartotojas gauna pranešimą apie negalimą šalinimą ir grąžinamas į tvarkaraščio formą. Taikomosios aplikacijos architektūros schema bei pilnas varotojo vadovas pateikti 1 ir 2 prieduose.

### 3. UNIVERSALIOS SAŠAJOS APLIKACIJOS KŪRIMO EKSPERIMENTINIS TYRIMAS

Universalios sąsajos aplikacijos kūrimas – tai galimybė sukurti vieną aplikaciją, kuri veiktų bet kuriame mobiliame įrenginyje nepriklausomai nuo jo naudojamo operacinio konteksto (OC) bei techninių parametrų. Operacinis kontekstas yra išorinė aplikacijos aplinka, kuri įtakoja jos funkcionavimą. Universalios sąsajos aplikacijos kūrimas reikalingas dėl operacinių kontekstų įvairovės, kurių klasifikavimas yra toks [8]:

- Techninės įrangos įvairovė – skirtingi ekrano parametrai, atminties dydis, procesorius, įvedimo būdas, papildoma įranga (kamera, balso įrašymas) bei sujungimo galimybės.
- Programinės įrangos įvairovė – skirtingos platformos, API standartai, privatūs API, permainingi priėjimas prie techninės įrangos, skirtingas daugialypės terpės palaikymas, maksimalus leidžiamas atminties dydis bei programavimo klaidos ir gudrybės.
- Vartotojų nustatymų įvairovė – skirtingas kalbos, stiliaus arba pasiekiamumo palaikymas.
- Aplinkos įvairovė – infrastruktūros išdėstymo įvairovė (įėjimo/išėjimo charakteristikos, atidarytos jungtys (angl. ports), prisijungimo prie išorinio tinklo nustatymai).

Kaip matome iš šio klasifikavimo operaciniai kontekstai tarpusavyje gali skirtis daugeliu faktorių, be to dauguma jų turi tam tikros įtakos kuriant naują aplikaciją. Universalios sąsajos aplikacijos kūrimas apima visas mobilios programinės įrangos projekto disciplinas [8]:

- Modeliavimas: analizė turi nustatyti optimalų operacinių kontekstų skaičių, kuriuose veikia aplikacija bei kurią rinkos dalį apima pasirinkti operaciniai kontekstai.
- Reikalavimų valdymas: sąveika tarp vartotojo ir aplikacijos skiriasi priklausomai nuo operacinio konteksto. Analitikui reikia gilių žinių apie kiekvieną iš operacinių kontekstų.
- Įgyvendinimas: programuotojas turi pritaikyti aplikaciją visiems nusistatytiems operaciniam kontekstam.
- Testavimas: aplikacija išbandoma pasirinktuose operaciniuose kontekstuose.
- Projekto valdymas: universalios sąsajos aplikacijos kūrimas tęsiamas atsiradus naujiems OC.

Tai padidina produkto kokybę bei pašalina kliūtis patenkant į mobilių aplikacijų rinką, tačiau universalios sąsajos kūrimas padidina aplikacijos kūrimo kaštus ir užima daugiau laiko.

Universalios sąsajos aplikacijos kūrimas galimas trim metodais. Pats paprasčiausias universalios sąsajos aplikacijos kūrimo būdas yra skirtingų versijų kūrimas, kurios veikia skirtinguose operaciniuose kontekstuose. Šios versijos didžia dalimi bus panašios, tačiau skirtingos subtiliose vietose, ten kur to reikalauja tam tikras operacinis kontekstas. Kuriant aplikacijas šiuo būdu dažniausiai

naudojama kopijavimo ir įterpimo technika. Skirtingų versijų kūrimo būdas padvigubina darbo, be to daugeliu aspektų išauga programinės įrangos kūrimo pastangos (pvz. klaidų taisymas šimte skirtingų versijų). Toliau aptarsiu du alternatyviu būdus, kuriais galima sumažinti tokias pastangas.[8]

Specifinio operacinio konteksto versijos nustatymas iš vienos programinio kodo bazės. Programuotojas parašo specialias instrukcijas tam tikam operaciniam kontekstui atskirai nuo aplikacijos programinio kodo. Suformuotas skriptas sukuria vieną versiją kiekvienam operaciniam kontekstui, sujungdamas nustatytas operacinio konteksto instrukcijas bei reikiamus taikomosios aplikacijos komponentus. Šiuo atveju taip pat turime skirtingas versijas kiekvienam operaciniam kontekstui, tačiau turime viena programinio kodo bazę, todėl reikia mažiau rankinio programavimo. Be to nebereikia dubliuoti to paties programinio kodo kopijų.[8]

Vienos aplikacijos versijos sukūrimas, kuri gali veikti skirtinguose operaciniuose kontekstuose. Kuriant universalios sąsajos aplikaciją šiuo būtu galima kurti aplikaciją kuri pritaikyta mažiausią ekraną turinčiam įrenginiui ir palaikanti paprasčiausias funkcijas, tačiau tai būtų neefektyvu technologiniu aspektu. Todėl programinė įranga kuriama taip, kad jos paleidimo metu ji galėtų identifikuoti operacinį kontekstą ir prisitaikyti prie jo reikalavimų. Paleidimo metu prie operacinio konteksto prisitaikančios aplikacijos užima daugiau vietos ir užtrunka daugiau laiko jas paleidžiant.[8]

Vairavimo mokyklos aplikacijos universalios sąsajos kūrimui pasirinktas vienos versijos aplikacijos sukūrimo metodas. Šis metodas pasirinktas, todėl, kad šiuo būdu aplikaciją sudaro viena versija, kuri gali veikti bet kuriame mobiliame įrenginyje jei jis numatytas aplikacijos programiniame kode. Be to ši aplikacija nėra imli resursams, todėl ją pritaikant skirtingiems operaciniams kontekstams resursų poreikis neišaug.

### **3.1. Eksperimentinis tyrimas**

Atliekant tiriamąjį darbą taikomoji aplikacija buvo įdiegta skirtingose mobilių įrenginių platformose ir praktiškai išbandytos aplikacijos funkcijos. Pasirinktas tiriamasis darbas yra aktualus, nes, kaip minėta, yra pagaminta įvairių mobilių įrenginių, kurie turi skirtingus operacinius kontekstus ir nėra tarpusavyje suderinti. Dėl šios priežasties taikomoji aplikacija skirtingai veikia skirtingo modelio mobiliuose įrenginiuose.

Atliekant bandymus pastebėti sutrikimai, trūkumai bei privalumai. Pagal taikomosios aplikacijos funkcijas jie surašyti žemiau esančiose lentelėse. Naudojantis atlikto tyrimo rezultatais atnaujinta taikomoji aplikacija ir pritaikyta skirtingiems mobiliems įrenginiams.

**3 lentelė. Taikomosios aplikacijos testavimo rezultatai**

	<b>S40 2nd Editon</b>	<b>S40 3rd Editon</b>	<b>S40 5th Editon</b>	<b>S40 6th Editon</b>
<b>Aplikacijos įdiegimas</b>	Atsiunčiant iš kito įrenginio per bluetooth sąsają arba parsisiunčiant iš interneto.	Atsiunčiant iš kito įrenginio per bluetooth sąsają arba parsisiunčiant iš interneto.	Sujungus telefoną su kompiuterių per USB jungtį arba parsisiunčiant iš interneto.	Sujungus telefoną su kompiuterių per USB jungtį arba parsisiunčiant iš interneto.
<b>Datos įvedimas</b>	Datos įvedimas iš klaviatūros naujai atsidariusiame lange, nors numatytas pasirinkimas iš kalendoriaus.	Datos įvedimas iš klaviatūros, nors numatytas pasirinkimas iš kalendoriaus.	Datos įvedimas iš klaviatūros, nors numatytas pasirinkimas iš kalendoriaus.	Datos įvedimas iš klaviatūros, nors numatytas pasirinkimas iš kalendoriaus.
<b>Naujas įrašas</b>	Naujai įvestas įrašas pasislepia po telefone nustatyto fono, įrašas matomas iš naujo įjungus aplikaciją. Atnaujinta aplikacija, t.y. lentelės generavimo vieta.	Naujai įvestas įrašas pasislepia po telefone nustatyto fono, įrašas matomas iš naujo įjungus aplikaciją. Atnaujinta aplikacija, t.y. lentelės generavimo vieta.	Naujai įvestas įrašas pasislepia po telefone nustatyto fono, įrašas matomas iš naujo įjungus aplikaciją. Atnaujinta aplikacija, t.y. lentelės generavimo vieta.	Naujai įvestas įrašas pasislepia po telefone nustatyto fono, įrašas matomas iš naujo įjungus aplikaciją. Atnaujinta aplikacija, t.y. lentelės generavimo vieta.
<b>Įrašo šalinimas ir redagavimas</b>	Veikia kaip numatyta	Veikia kaip numatyta	Veikia kaip numatyta	Veikia kaip numatyta
<b>Padalinio, kliento, instruktoriaus ir automobilio pasirinkimas</b>	Pasirinkimas įvedamas iš naujai atsidariusio lango. Patvirtinimo mygtukas paslėptas po options mygtuku, todėl reikia atlikti papildomą veiksmą.	Pasirinkimas įvedamas iš naujai atsidariusio lango, tačiau tai nepatogumų nesukelia.	Pasirinkimas įvedamas iš naujai atsidariusio lango, tačiau tai nepatogumų nesukelia.	Veikia kaip numatyta
<b>Duomenų atvaizdavimas</b>	128x160 rezoliucijos ekrane telpa tik 1 lentelės stulpeliai iš 5.	128x160 rezoliucijos ekrane telpa tik 1 lentelės stulpeliai iš 5.	128x160 rezoliucijos ekrane telpa tik 1 lentelės stulpeliai iš 5.	320x480 rezoliucijos ekrane telpa 4 lentelės stulpeliai iš 5.

**Šaltinis:** sudaryta autoriaus

S40 programinės įrangos platforma naudojama mobiliuosiuose įrenginiuose, kurią pagamino Nokia. Tai plačiausiai naudojama platforma kuri veikia milijonuose telefonų. Ji geba atlikti dažniausiai sutinkančias telefonuose funkcijas, kaip kameros valdymą, interneto naršymą, bluetooth ryšį, naujausios kartos telefoniją. S40 į savo sistemos kodą neleidžianti įsiterpti trečiosioms programoms ir nesugebanti su jomis tvarkytis. Dar ji pasižymi tuo, kad palaiko Flash failus. Šiuo metu naujausia S40 versija yra 6, kurios interneto naršymas labai sparčiai artėja prie asmeninių kompiuterių.

Taikomosios aplikacijos funkcijos, kurios veikia kaip numatyta arba ištaisytos atradus neatitikimus:

- Mobilūs įrenginiai su pirma arba antra S40 platformos versija nepalaiko sujungimo su kompiuteriu per USB jungtį, tačiau aplikaciją įdiegti galima persiuntus ją kitu įrenginiu arba parsisiuntus internetu (žr. 2 lentelė, 1 eilutė).
- Įvedant naują įrašą jis pasislepia po nustatytu ekrano fonu. Tai svarbiausiai problema atrasta atliekant ši eksperimentinį tyrimą. Ją pavyko išspręsti atnaujinus taikomosios aplikacijos programinį kodą (žr. 2 lentelė, 3 eilutė).
- Įrašo šalinimo ir redagavimo funkcija veikė be sutrikimų visuose bandytuose S40 platformos versijose (žr. 2 lentelė 4 eilutę).

Taikomosios aplikacijos funkcijos, kurios neveikia taip kaip numatyta, tačiau atlieka joms numatytas užduotis:

- Visos S40 platformos versijos neleidžia pasirinkti datos iš kalendoriaus, ją reikia įvesti klaviatūra. Šios problemos išspręsti negalima, nes S40 platformos funkcijų bibliotekose nėra tokios funkcijos (žr. 2 lentelė, 2 eilutė).
- Duomenų pasirinkimas iš sąrašo veikė kaip numatyta tik šeštoje platformos versijoje. Ankstesnėse versijose duomenys pasirenkami iš naujame lange atsidarančio sąrašo. Kadangi senesnės S40 platformos versijos neturi iššokančio sąrašo galimybės, todėl problemos išspręsti negalima (žr. 2 lentelė, 5 eilutė).
- Šešta S40 platformos versija geriausiai pateikia duomenis lentelėje, nes šios platformos mobilūs įrenginiai turi 320 x 480 rezoliucijos ekraną. Senesnės versijos platformose atvaizduojama tik 1 lentelės stulpelis iš 5 ir tai apsunkina duomenų peržiūrą (žr. 2 lentelė, 6 eilutė).

**4 lentelė. Taikomosios aplikacijos testavimo rezultatai**

	<b>S60 2nd edition</b>	<b>S60 3rd edition</b>	<b>S60 5th edition</b>
<b>Aplikacijos įdiegimas</b>	Atsiunčiant iš kito įrenginio per bluetooth sąsają arba parsisiunčiant iš interneto.	Sujungus telefoną su kompiuterių per USB jungtį arba parsisiunčiant iš interneto.	Sujungus telefoną su kompiuterių per USB jungtį arba parsisiunčiant iš interneto.
<b>Datos įvedimas</b>	Datos įvedimas iš klaviatūros, nors numatytas pasirinkimas iš kalendoriaus.	Datos įvedimas iš klaviatūros, nors numatytas pasirinkimas iš kalendoriaus.	Datos įvedimas iš klaviatūros, nors numatytas pasirinkimas iš kalendoriaus.
<b>Naujas įrašas</b>	Naujai įvestas įrašas pasislepia po telefone nustatyto fono, įrašas matomas iš naujo įjungus aplikaciją. Atnaujinta aplikacija, t.y. lentelės generavimo vieta.	Naujai įvestas įrašas pasislepia po telefone nustatyto fono, įrašas matomas iš naujo įjungus aplikaciją. Atnaujinta aplikacija, t.y. lentelės generavimo vieta.	Naujai įvestas įrašas pasislepia po telefone nustatyto fono, įrašas matomas iš naujo įjungus aplikaciją. Atnaujinta aplikacija, t.y. lentelės generavimo vieta.
<b>Įrašo šalinimas ir redagavimas</b>	Veikia kaip numatyta	Veikia kaip numatyta	Veikia kaip numatyta
<b>Padalinio, kliento, instruktoriaus ir automobilio pasirinkimas</b>	Pasirinkimas įvedamas iš naujai atsidariusio lango. Patvirtinimo mygtukas paslėptas po „options“ mygtuku, todėl reikia atlikti papildomą veiksmą.	Veikia kaip numatyta.	Veikia kaip numatyta.
<b>Duomenų atvaizdavimas</b>	128x160 rezoliucijos ekrane telpa tik 2 lentelės stulpeliai iš 5.	Ekranų rezoliucija 240x320, todėl telpa 3 lentelės stulpeliai iš 5.	240x320 rezoliucijos ekrane telpa 3 lentelės stulpeliai iš 5. Lentelės duomenys atvaizduojami šviesiu šriftu, todėl buvo atnaujinta atvaizdavimo spalva.

**Šaltinis:** sudaryta autoriaus

S60 yra programinės įrangos platforma mobiliesiems įrenginiams, kuri veikia Symbian operacine sistema. S60 yra pirmaujanti platforma pasaulyje skirta išmaniesiems telefonams. Ją išvystė Nokia ir suteikė patentus kitoms kompanijoms kaip: Lenovo, LG Electronics, Panasonic ir Samsung. S60 standarto dėka telefonai gali veikti su didelės raiškos ekranais, palaikyti daugelį galingų žaidimų ir programų, kurias galima papildyti bei atnaujinti. Su S60 platforma vienu metu galima paleisti kelias programas ir nereikia nerimauti, kad nutrūks šiuo metu pradėta užduotis.

Taikomosios aplikacijos funkcijos, kurios veikia kaip numatyta arba ištaisytos atradus neatitikimus:

- S60 2nd Edition platformos mobilieji įrenginiai neturi sujungimo galimybės su kompiuteriu per USB jungtį, tačiau naujesnės platformos versijos pilnai išpildo šia funkciją (žr. 3 lentelė, 1 eilutė).
- Įvedant naują įrašą jis pasislepia po nustatytu ekrano fonu. Tai svarbiausiai problema atrasta atliekant ši eksperimentinį tyrimą. Ją pavyko išspręsti atnaujinus taikomosios aplikacijos programinį kodą (žr. 3 lentelė, 3 eilutė).
- Įrašo šalinimo ir redagavimo funkcija veikė be sutrikimų visuose bandytuose mobiliuose įrenginiuose (žr. 3 lentelė 4 eilutė).
- Skirtingai nuo S40 platformos S60 platforma pilnai palaiko iššokančio sąrašo funkciją. Mobiluose įrenginiuose su šia platforma vairavimo mokyklos padalinį, mokinių, instruktorių bei automobilį galima pasirinkti iš iššokančio sąrašo (žr. 3 lentelė, 5 eilutė).

Taikomosios aplikacijos funkcijos, kurios neveikia taip kaip numatyta, tačiau atlieka joms numatytas užduotis:

- Visose S60 platformose versijose, taip kaip ir S40 platformoje, negalimas datos pasirinkimas iš kalendoriaus, ją reikia įvesti klaviatūra. Šios problemos išspręsti negalima, nes S60 platformos funkcijų bibliotekose nėra tokios funkcijos (žr. 3 lentelė, 2 eilutė).
- S60 platformos trečia ir penkta versijos yra pritaikytos veikti mobiliuose įrenginiuose su didesnės rezoliucijos ekranais (nuo 240x320). Šiuose įrenginiuose yra atvaizduojami 3 iš 5 vairavimo tvarkaraščio lentelės stulpeliai (žr. 3 lentelė, 6 eilutė).



**5 lentelė. Taikomosios aplikacijos testavimo rezultatai**

	<b>Samsung SDK 1.1</b>	<b>Sony-Ericsson SDK 2.5</b>	<b>Motorola SDK for Java ME</b>
<b>Aplikacijos įdiegimas</b>	Parsisiunčiant iš interneto arba kito įrenginio per IR sąsają	Sujungus telefoną su kompiuterių per USB jungtį arba parsisiunčiant iš interneto.	Sujungus telefoną su kompiuterių per USB jungtį arba parsisiunčiant iš interneto.
<b>Datos įvedimas</b>	Veikia kaip numatyta.	Veikia kaip numatyta.	Veikia kaip numatyta.
<b>Naujas įrašas</b>	Veikia kaip numatyta.	Veikia kaip numatyta	Naujai įvestas įrašas pasislepia po telefone nustatyto fono, įrašas matomas iš naujo įjungus aplikaciją. Atnaujinta aplikacija, t.y. lentelės generavimo vieta.
<b>Įrašo šalinimas ir redagavimas</b>	Veikia kaip numatyta	Veikia kaip numatyta	Veikia kaip numatyta
<b>Padalinio, kliento, instruktoriaus ir automobilio pasirinkimas</b>	Veikia kaip numatyta.	Veikia kaip numatyta	Veikia kaip numatyta, tačiau patvirtinimo mygtukas paslėptas po „options“ mygtuku, todėl reikia atlikti papildomą veiksmą.
<b>Duomenų atvaizdavimas</b>	Ekrane telpa tik 2 lentelės stulpeliai iš 5. Telefonas turi tik 128x160 dydžio rezoliucija, todėl yra nepajėgus atvaizduoti daugiau duomenų viename ekrane.	Ekrane telpa tik 2 lentelės stulpeliai iš 5. Telefonas turi tik 128x160 dydžio rezoliucija, todėl yra nepajėgus atvaizduoti daugiau duomenų viename ekrane.	Įrenginys su 320x240 rezoliucijos ekranu atvaizduoja visus lentelės stulpelius, o mažesni ekranai vaizduoja 3 iš 5 stulpelių.

**Šaltinis:** sudaryta autoriaus

Šioje dalyje nagrinėtos mažiau paplitusios mobilių įrenginių platformos Samsung SDK 1.1, Sony-Ericsson SDK 2.5 bei Motorola SDK for Java ME. Samsung SDK 1.1 platforma yra skirta CLDC ir MIDP tipo Java konfigūracijoms, tuo tarpu Sony-Ericsson ir Motorola platformos skirtos CDC konfigūracijai. Platformų emuliatoriai turi emuliacijos aplinką, derinimo (debug) ir optimizavimo įrankius bei keletą demonstracinių aplikacijų.

Taikomosios aplikacijos funkcijos, kurios veikia kaip numatyta arba ištaisytos atradus neatitikimus:

- Samsung SDK 1.1 platformos mobilūs įrenginiai neturi sujungimo su kompiuteriu per USB jungtį. Kitų dviejų platformų mobiliuose įrenginiuose aplikacijos gali būti įdiegtos iš kompiuterio arba atsiuntus per bevielius ryšius (žr. 4 lentelė, 1 eilutė).
- Šios telefonų platformos turi naudojamą kalendoriaus bibliotekas, todėl visuose modeliuose galimas datos pasirinkimas iš kalendoriaus (žr. 4 lentelė, 2 eilutė).
- Motorola SDK for Java ME platformoje naujai įterpiamas įrašas pasislepia po nustatyto ekrano fonu. Ši problema išspręsta atnaujinus taikomosios aplikacijos programinį kodą. Tuo tarpu Samsung ir Sony-Ericsson platformos šią funkciją įvykdo taip kaip numatyta. (žr. 4 lentelė, 3 eilutė).
- Įrašo šalinimo ir radagavimo funkcija be priekaištų veikė visuose bandytuose mobiliuose įrenginiuose (žr. 4 lentelė, 4 eilutė).
- Duomenų parinkimas iš sąrašo veikė kaip numatyta visose bandytose mobilių įrenginių platformose, tačiau Motorola platformoje pasirinkimo patvirtinimo mygtukas yra paslėptas po „options“ mygtuku todėl vartotojas privalo atlikti viena papildomą veiksmą (žr. 4 lentelė, 5 eilutė).

Taikomosios aplikacijos funkcijos, kurios neveikia taip kaip numatyta, tačiau atlieka joms numatytas užduotis:

- Motorola platformos mobilūs įrenginiai su 320x240 rezoliucijos ekranu yra pajėgūs atvaizduoti visus lentelės stulpelius, o tos pačios platformos įrenginiai su 240x320 ekranu atvaizduoja 3 iš 5 stulpelių. Kitų dviejų platformų ekranų rezoliucijos yra mažesnės, todėl juose vaizduojami tik 2 iš 5 stulpelių (žr. 4 lentelė, 6 eilutė).

### **3.2. Eksperimentinio tyrimo apibendrinimas**

Atliktas tyrimas leido pastebėti du pagrindinius aplikacijos trūkumus, tai įvestų įrašų pasislėpimas po nustatyto fonu ir duomenų atvaizdavimo problema.

- S40 ir S60 platformose buvo pastebėta, kad įvedant naują įrašą jis pasislepia po nustatyto įrenginio fonu, tačiau šią problemą pavyko išspręsti atnaujinus programinį kodą. Nauji įrašai teisingai buvo įvedami senesnio tipo platformose, nes šios dažnai neturi ekrano fono.
- Motorolos platformos telefonai su 320x240 ekrano rezoliucija atvaizduoja visus lentelės duomenis viename ekrane, tačiau kitų platformų įrenginiai nesugeba atvaizduoti pilnos duomenų lentelės dėl ribotos rezoliucijos. Trūkumas gali būti išspręstas tik mobiliuose

įrenginiuose kurių ekrano rezoliucija yra 240x320 arba didesnė. Tokiuose įrenginiuose siūloma duomenų lentelę atvaizduoti horizontaliai. Šiuo atveju matytume visus lentelės stulpelius viename ekrane. Tokie pakeitimai reikalauja esminių aplikacijos pakeitimų, be to gali iškilti suderinamumo problemų su mobiliais įrenginiais kurių ekrano rezoliucija mažesnė nei 240x320 taškų.

Dauguma mobilių įrenginių platformų neturi datos įvedimo iš kalendoriaus galimybės, todėl data įvedama įrenginio klaviatūra. Be to atlikus eksperimentinį tyrimą pastebėta, kad datos pasirinkimas iš kalendoriaus užtrunka ilgiau nei įvedimas klaviatūra. Dėl šių priežasčių galima atsisakyti kalendoriaus funkcijos, o palikti tik paprastą datos įvedimą. Tai atlikus bus sutaupyta truputį mobilios įrenginio atminties resursų.

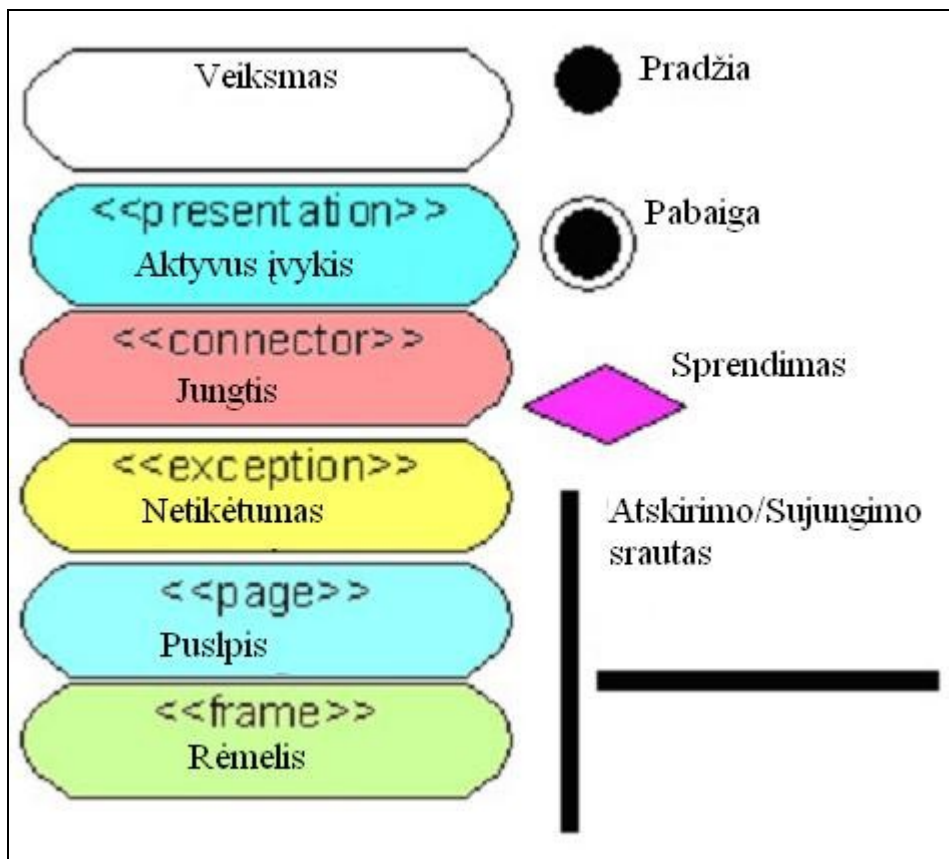
### **3.3. Pasiūlymai**

Kuriant universalios sąsajos taikomasias aplikacijas kūrėjai išnaudoja daug žmogiškųjų resursų testuodami savo kuriamą sistemą. Kuriant programas siekiama testuoti kuo platesnėje mobilių telefonų aibėje, o testavimą atlieka žmogiškieji resursai. Toks testavimas užima daug laiko, be to jis yra mažiau patikimas klaidų atžvilgiu nei automatinis testavimas. Šiuo metu egzistuoja labai mažai automatinio testavimo priemonių, veikiančių skirtinguose mobiliuose įrenginiuose, skirtų mobilių sistemų vartotojo sąsajai testuoti. Egzistuojančios priemonės paprastai gali būti naudojamos tik kaip įrašymo patikrinimo priemonės.

Testų generavimas naudojant UML diagramas yra mažai žinomas būdas, tačiau tai yra siūloma naudoti varotojo sąsajai modeliuoti. Grafiniai elementai gali būti langai, dialogai, konteineriai, mygtukai, įvedimo langai ir pan. Be grafinių elementų vartotojo sąsajoje svarbią vietą užima navigaciniai veiksmai, t.y. įvairaus tipo jungiamieji elementai. Taigi grafinę vartotojo sąsają galima apibrėžti aibe grafinių elementų ir aibe navigacinių veiksmų. Tai yra optimalus būdas sistemos projektuotojui atvaizduoti grafinę vartotojo sąsają diagramos pavidalu. Reikia tik apibrėžti diagramos elementus anksčiau minėtoms aibėms atvaizduoti. Siūloma naudoti UML veiklos diagramas su žemiau pateiktais stereotipais [10]:

- „Action“ (veiksmas) – pirminis diagramos elementas. Tai veiksmas netiesiogiai iššauktas sistemos ar vartotojo. Kadangi tai yra pats paprasčiausias elementas jis žymimas balta spalva arba nespalvinamas.
- „Presentation“ (aktyvus įvykis) – vartotojo aktyvuojamas elementas. Tai paprastai yra tarpinė grandis tarp panaudos atvejo aktoriaus ir sistemos. Šis stereotipas skiriasi nuo veiksmo tuo, kad jis yra tiesiogiai prieinamas vartotojui.

- „Connector“ (jungtis) – vartotojo veiksmas, kuris iššaukia išorės sistemos ar posistemės įvykį.
- „Exception“ (netikėtumas) – diagramos šaka, kuri įvykdoma tik atsitikus nenumatytai sistemos klaidai ar atsiradus nenumatytai sistemos būsenai. Šis elementas parodo sistemos prisijungimo ir atsistatymo būsenos veiksmus.
- „Page“ (puslapis) – jis yra naudojamas atskirti neapjungiamus ekrano vaizdus.
- „Frame“ (rėmelis) – rėmelis atskiria logines ar fizines grafinės vartotojo sąsajos dalis. Gali būti naudojamas keliuose puslapiuose.



Šaltinis: <http://www.ibm.com/developerworks/rational/library/4697.html>

12 pav. Veiklos diagramos elementai.

Testavimas būtų atliekamas testavimo programai imituojant vykdymo sąrašę pateiktus veiksmus, tikrinant varotojo sąsajos struktūras kiekviename žingsnyje ir lyginant jas su laukiamomis struktūromis. Šioms struktūroms nesutampant pranešimas apie rastą klaidą yra išsaugomas atmintyje ir testavimas vykdomas toliau kol pasiekama veiksmų sąrašo pabaiga.

## IŠVADOS

1. Išanalizuotos Eclipse IDE, NetBeans IDE, Python S60 ir IntelliJ IDEA programavimo įrankių galimybės, privalumai ir trūkumai. Pagal gautus rezultatus, praktiniam galimybių panaudojimui pasirinktas NetBeans IDE programavimo įrankis.
2. Suformuotas vairavimo mokyklos informacinės sistemos konceptinis modelis bei suprojektuota mobili informacinės sistemos versija. Šie darbai atlikti naudojantis Microsoft Visio įrankiu.
3. Sukurta taikomoji aplikacija skirtą vairavimo mokyklos darbuotojų veiklai optimizuoti, be to naudojama eksperimentiniams tyrimams atlikti siekiant ją pritaikyti skirtingiems mobiliems įrenginiams. Aplikacijos kūrimui naudotas NetBeans IDE 6.5 programavimo įrankis. Prieduose pateiktas taikomosios aplikacijos vartotojo vadovas.
4. Atliktas taikomosios aplikacijos eksperimentinis tyrimas. Nustatyta, kad taikomoji aplikacija blogiausiai veikė S40 platformos įrenginiuose, tuo tarpu Samsung ir Sony-Ericsson platformos atlieka daugumą funkcijų taip kaip numatyta.
5. Atnaujintas taikomosios aplikacijos duomenų atvaizdavimo lentelės generavimas, todėl naujai įvedami įrašai nepasislepia po nustatytų įrenginio ekrano fonu, be to atsisakyta datos įvedimo iš kalendoriaus funkcijos.
6. Nustatyta, kad ateityje duomenų vaizdavimui turėtų būti pasirinkta horizontali lentelė. Šiuo būdu galima matyti visus duomenis viename lange. Taip pat pasiūlyta universalios sąsajos testavimui naudoti UML diagramomis paremtą programinę įrangą.

## LITERATŪRA

1. Java Micro Edition (ME) [Paskutinį kartą žiūrėta 2009.01.13]. Prieiga Internetu:  
<<http://java.sun.com/javame/index.jsp>>
2. Wikipedia Free Encyclopedia [Paskutinį kartą žiūrėta 2009.01.13]. Prieiga Internetu:  
<[http://en.wikipedia.org/wiki/Java\\_ME](http://en.wikipedia.org/wiki/Java_ME)>
3. Python ir Python for S60 programavimo kalbos [Paskutinį kartą žiūrėta 2009.01.13]. Prieiga Internetu: <<http://www.python.org/> >
4. NetBeans IDE programavimo įrankis. [Paskutinį kartą žiūrėta 2009.01.17]. Prieiga internetu  
<<http://www.netbeans.org/features/index.html> >
5. Eclipse Platform Technical Overview. [Paskutinį kartą žiūrėta 2009.05.03]. Prieiga internetu  
<<http://www.eclipse.org/whitepapers/eclipse-overview.pdf>>
6. IntelliJ IDEA Features & Screenshots. [Paskutinį kartą žiūrėta 2009.05.03]. Prieiga internetu  
<<http://www.jetbrains.com/idea/features/index.html>>
7. Java Servlet aprašymas [Paskutinį kartą žiūrėta 2009.01.17] Prieiga internetu  
<[http://lt.wikipedia.org/wiki/Java\\_Servlet](http://lt.wikipedia.org/wiki/Java_Servlet)>
8. Fragmentation of mobile application [Paskutinį kartą žiūrėta 2009.03.01]. Prieiga Internetu:  
<<http://www.comp.nus.edu.sg/~damithch/df/device-fragmentation.htm>>
9. UML Activity Diagrams: Detailling User Interface Navigation [Žiūrėta 2009.05.02]. Prieiga internetu <<http://www.ibm.com/developerworks/rational/library/4697.html>>
10. JavaWorld. Eclipse or NetBeans. [Žiūrėta 2009.05.02]. Prieiga internetu  
<<http://www.javaworld.com/javaworld/jw-03-2008/jw-03-java-ides0308.html?page=2> >
11. Eclipse Platform Technical Overview. [Žiūrėta 2009.05.02]. Prieiga internetu  
<<http://www.eclipse.org/whitepapers/eclipse-overview.pdf> >
12. Accessible development for S60 devices. [Žiūrėta 2009.05.02]. Prieiga internetu  
<[http://www.forum.nokia.com/Resources\\_and\\_Information/Tools/Runtimes/Python\\_for\\_S60/](http://www.forum.nokia.com/Resources_and_Information/Tools/Runtimes/Python_for_S60/)>
13. Resolving Java ME Device Fragmentation Issues Using NetBeans. [Žiūrėta 2009.05.02].  
Prieiga internetu <<http://www.netbeans.org/kb/60/mobility/javame-devicefragmentation.html> >
14. Java Editors Comparison. [Žiūrėta 2009.05.02]. Prieiga internetu  
<<http://www.scribd.com/doc/11758258/Java-Editors-Comparison-NetBeans-IDE-Eclipse-IntelliJ-IDEA>>
15. Eclipse Java2ME Plugin Documentation. [Žiūrėta 2009.05.02]. Prieiga internetu  
<<http://eclipseme.org/docs/index.html>>

16. JavaWorld. The Fragmentation effect. [Žiūrėta 2009.05.02]. Prieiga internetu  
<<http://www.javaworld.com/javaworld/jw-05-2004/jw-0524-fragment.html?page=1>>
17. WAMP server, Apache, PHP, MySQL on Windows. [Žiūrėta 2008.11.02]. Prieiga internetu  
<<http://www.wampserver.com/en/>>
18. IntelliJ IDEA Licensing & Upgrade. [Žiūrėta 2009.05.02]. Prieiga internetu  
<<http://www.jetbrains.com/idea/buy/index.jsp>>
19. Jurgen Scheible, Ville Tuulos. Mobile Python. West Sussex PO19 8SQ, England, 2007, p. 30.
20. Ingrida Šarkiūnaitė, Dalia Krikščiūnienė, Rimvydas Simutis. MAGISTRO BAIGIAMOJO DARBO RENGIMO TVARKA. Metodiniai nurodymai. Kaunas, 2007
21. Tadas Lingė. Universalios sąsajos aplikacijų kūrimas mobiliems įrenginiams. Straipsnis. 14-oje tarpuniversitetinė magistrantų doktorantų mokslinė konferencija. Kaunas, 2009 gegužės 8 d.



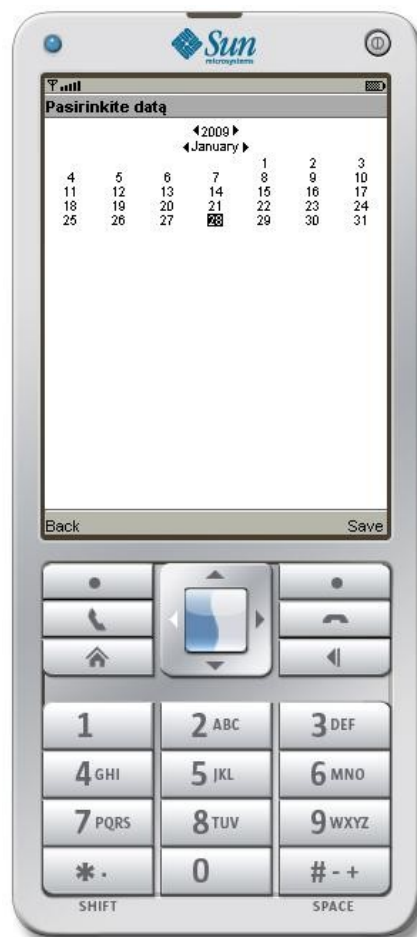


## Vartotojo vadovas

Aplikacijos failai TestGUI.jad ir TestGUI.jar įkeliami į mobilų įranginį. Ji paleidžiama pasirinkus TestGUI.jar failą. TestGUI.jad faile saugoma meta informacija. Paleidus aplikaciją matysime vaizdą pateiktą žemiau esančiame paveiksle.

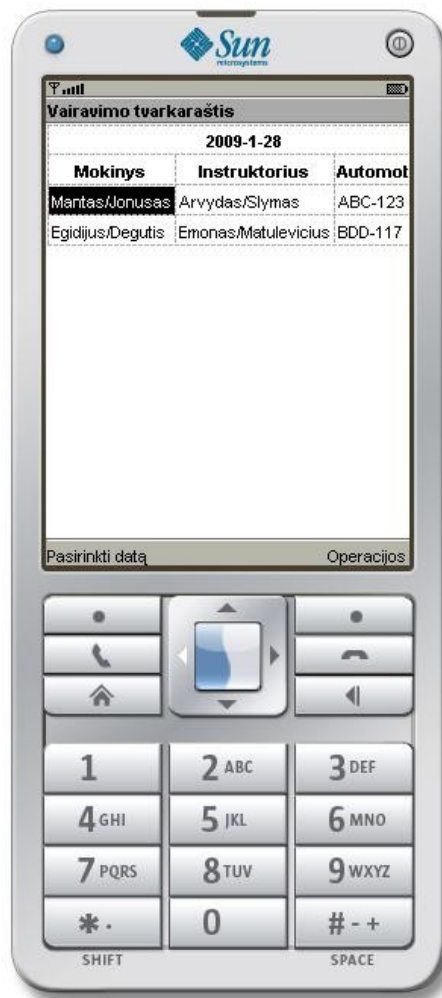


Šaltinis: sudaryta autoriaus  
10 pav. Datos ir miesto pasirinkimas.



Šaltinis: sudaryta autoriaus  
11 pav. Datos pasirinkimas

Pasirinkus praktinio vairavimo miestą ir datą bus atvaizduojami duomenys kurie pavaizduoti žemiau esančiame paveiksle.



Šaltinis: sudaryta autoriaus  
12 pav. Praktinio vairavimo duomenų atvaizdavimas

Paveiksle vaizduojami praktinio vairavimo duomenys:

- Mokinio vardas ir pavardė
- Vairavimo instruktoriaus vardas ir pavardė
- Automobilio numeris
- Praktinio vairavimo pradžios laikas
- Praktinio vairavimo pabaigos laikas

Telefono emulatoriaus ekranas nepalaiko reikiamos rezoliucijos, todėl ekrane netelpa visi duomenys. Ekrane netelpančius duomenis galima peržiūrėti mobilaus įrenginio klaviatūros pagalba vaikstant po lentelę. Eilutė, kurioje yra pažymėta celė yra aktyvi, todėl su ja gali būti vykdomi tolesni veiksmai. Menu pasirinkus „Operacijos“ atidaromas žemiau esantis langas, kuriame galima pasirinkti reikiama operacija.



Šaltinis: sudaryta autoriaus.

13 pav. Operacijos pasirinkimo langas.

Aktyvi eilutė gali būti šalinama arba atnaujinama jei įrašo data nėra vėlesnė nei šiandienos.

Naujas įrašas gali būti įkeliamas tik naujesnei dienai nei ši diena.



Šaltinis: sudaryta autoriaus.

14 pav. Naujo įrašo įvedimas



Šaltinis: sudaryta autoriaus.

15 pav. Laiko pasirinkimas

Mokyns, vairavimo instruktorius bei automobilio numeris pasirenkami iš sąrašo. Praktinio vairavimo pradžios ir pabaigos laikas pasirenkami mobilaus įrenginio klaviatūra. Laiko pasirinkimo vaizdas pateiktas 9 paveiksle.

# Pranešimo medžiaga. Universalios sąsajos aplikacijų kūrimas mobiliems įrenginiams

## UNIVERSALIOS SĄSAJOS APLIKACIJŲ KŪRIMAS MOBILIEMS ĮRENGINIAMS

Tadas Lingė

*Vilniaus universitetas, Kauno humanitarinis fakultetas, Muitinės 8, Kaunas, Lietuva*

Straipsnyje nagrinėjamos mobilių įrenginių programavimo priemonės NetBeans ir Eclipse. Išskiriami tarpusavio pranašumai ir trūkumai. Taip pat pristatoma taikomoji aplikacija sukurta naudojantis NetBeans programavimo įrankiu bei analizuojamas aplikacijos veikimas skirtinguose mobiliuose įrenginiuose.

Raktiniai žodžiai: mobilus įrenginys, mobilus telefonas, programavimas, JavaMe, NetBeans IDE Eclipse MTJ, universali sąsaja.

### 1. Įvadas

Mobilieji telefonai gali būti naudojami ne tik skambinti bei trumposioms žinutėms siųsti, bet ir keistis įvairiausiais duomenimis, siųsti kompiuterinėms programoms, naršyti internete, naudotis įvairiomis elektroninėmis paslaugomis. Dauguma interneto vartotojų nori šiomis galimybėmis naudotis mobiliuose aplinkoje, nepriklausomoje nuo buvimo vietos ir laiko.

Viskas pasikeitė rinkoje pasirodžius išmaniesiems telefonams, delniniams kompiuteriams bei jų programavimo priemonėms: Java ME, NetBeans, Eclipse, Visual Studio .NET, Python for S60 ir daugelis kitų. Šiais įrankiais pagaminta programinė įranga skirta mobiliems prietaisams, kurie sulaužė visus iki tol įmanomus apribojimus. Įvairūs žaidimai, žodynai, žemėlapiai, kelionių gidai, MPEG grotuvai ir kt. – visa tai šiandien gali turėti kiekvienas mobiliojo ryšio vartotojas, turintis tobulesnį telefoną.

Darbo uždaviniai:

- Mobilijų įrenginių programavimo priemonių apžvalga.
- Taikomosios aplikacijos kūrimas.
- Taikomosios aplikacijos adaptavimas įvairiems mobiliems įrenginiams.

### 2. Programavimo priemonės

Java IDE (Integruota programų kūrimo aplinka) rinkoje yra tik trys pagrindiniai konkurentai, tačiau konkurencija tarp jų ypač intensyvi. NetBeans ir Eclipse yra nemokami ir atviro kodo paketai, tuo tarpu IntelliJ yra mokamas programavimo įrankis. Toliau apžvelgsiu NetBeans 6.0 ir Eclipse 3.3, lyginant jų pagrindinius bruožus, kurie lemia IDE pasirinkimą. Abu IDE produktus yra paprasta atsisiųsti ir įdiegti. Taip pat gamintojų tinklapiuose galima atsisiųsti įvairių funkcijų konfigūracijas ir įskiepius (plugin), kurie reikalingi jūsų projektui. Šie programavimo įrankiai abi Java Me platformos konfigūracijas:

- Connected Limited Device Configuration (CLDC) yra konfigūracija skirta prietaisams su mažesne atminties talpa ir lėtesne duomenų apdorojimo sparta. Java ME dėl CLDC ir MIDP aplinkos, kurios yra daugelyje mobilių prietaisų, yra puikus įrankis MIDlet kūrimui. MIDlet yra taikomosios programos sukurtos Java ME programinės įrangos plėtotojų, tokie kaip žaidimai, verslo programos bei kitos mobilios ypatybės. Šios programos viena karta parašytos gali būti paleidžiamos visuose prietaisuose, kurie palaiko Java ME technologiją.
- Connected Device Configuration (CDC) – konfigūracija skirta prietaisams su daug didesne atminties talpa, didesne duomenų apdorojimo sparta ir tinklo galimybe. Tai išmanieji telefonai, TV priedai, komunikatoriai bei įmontuojami serveriai [1].

Eclipse programavimo aplinka (IDE) pasirinkta dėl universalumo ir plataus pritaikymo, bei paplitimo tarp vartotojų. Tai yra kelių kalbų programinės įrangos platforma, kurią sudaro IDE bei įskiepiai skirti sistemos praplėtimui.

Įrankis naudojamas Java aplikacijų kūrimui, be to įdiegus reikiamus įskiepius galima programuoti C/C++, Cobol, Python, Perl, PHP ir kitomis kalbomis.

Eclipse IDE turi Eclipse Mobile Tools for Java Project (MTJ) įskiepi, skirtą Java ME (Micro edition) aplikacijų kūrimui ir plėtojimui. Pagrindinės Eclipse IDE Mobile atliekamos funkcijos:

- J2ME MIDlet projektų kūrimas.
- J2ME MIDlet kūrimas.
- MIDlet paleidimas ir suderinimas.
- MIDlet pakavimas.
- Prietaisų valdymas.
- JAD ir JAR failų generavimas. [3]

NetBeans Mobility gali būti naudojamas aplikacijų, veikiančių plataus naudojimo mobiliuose prietaisuose su Java Micro Edition (Java ME) platforma, rašymui, testavimui ir suderinimui.

NetBeans Mobility įrankis apjungia CDC ir CLDC kūrimo savybes. NetBeans turi pilna programinės įrangos kūrimo ciklą, programinio kodo redagavimą, kompiliavimą, paleidimą bei klaidų paiešką. Šis paketas gali stabiliai veikti Linux, Mac bei Windows operacinėse sistemose.

NetBeans Mobility savybės:

- Projekto palaikymas, Apache Ant. Sutvarko MIDP programinį kodą ir projekto meta duomenis patogesniai valdymui. Taip pat leidžia kurti projektą už IDE ribų.
- Integracija su J2ME Wireless Toolkit 2.2.
- Prietaiso fragmentavimas – aplikacijos kūrimas sudėtiniais prietaisams ir vykdyti prietaiso specifinį kodą kaip konfigūraciją vienos aplikacijos viduje.
- MIDP lokalizacijų palaikymas – leidžia pridėti specifinius komponentus į kuriamą aplikaciją.
- J2ME MIDP 2.0 ir CLDC 1.1 standartų palaikymas
- Emuliatorių platformų palaikymas – aplikacijos testavimas įvairių prietaisų emulatoriuose, kuriuos teikia gamintojai.
- Integruotas optimizavimo palaikymas – pridedamas saugumas, pagerinamas veikimas ir aplikacijos dydžio optimizavimas
- Over-The-Air siuntimo testavimas – Emulatorius simuliuoja tikro mobiliojo įrenginio veikimą. [2]

NetBeans yra viena nuosekli priemonė, kuria galima parsisiųsti iš vieno šaltinio, tuo tarpu Eclipse yra sudaryta iš atskirų įskiepių, o tai komplikuoja įrankio įdiegimą. Taip pat NetBeans turi Tomcat serverį, jis bus reikalingas kuriant vairavimo mokyklos taikomąją aplikaciją. Netbeans ir Eclipse turi daug tokių pačių arba panašių funkcijų, todėl negalime teigti, kad kuris nors iš jų yra pranašesnis. Tolesniam tyrimo vystymui pasirinktas NetBeans paketas todėl, kad turi Tomcat serverį, o visas kitas reikalingas funkcijas turi abu įrankiai.

### 3. Taikomoji aplikacija

NetBeans programavimo įrankiu sukurta programinė įranga skirta vairavimo instruktoriams, kurie net neturėdami šalia kompiuterio gali patogiai tvarkyti praktinio vairavimo tvarkaraštį. Programinė įranga yra sudaryta iš dviejų dalių – serverinės dalies (servlet) ir aplikacijos. Servlet patalpintas įmonės serveryje, kartu su duomenų baze, o taikomoji aplikacija bus diegiama vartotojų (vairavimo instruktorių) mobiliuose įrenginiuose.

Mobiliame įrenginyje įdiegta taikomoji aplikacija turi šias funkcijas:

- Kliento registravimas praktiniam vairavimui.
- Užregistruoto praktinio vairavimo redagavimas.
- Užregistruoto praktinio vairavimo šalinimas.



**Paveikslas 1. Taikomoji aplikacija emuliacijoje**

Mobiliame įrenginyje esanti taikomoji aplikacija jungiasi prie serveryje esančio servlet ir per jį vartotojui pateikiami duomenys esantys duomenų bazėje. Mobiliaus įrenginio susijungimas su vairavimo mokyklos duomenų baze vyksta naudojant GPRS internetą. Mobiliame įrenginyje duomenys atvaizduojami lentelėje. Klaviatūros valdymo mygtukais vyksta vaikščiojimas po lentelės eiles, o aktyvią eilutę galima redaguoti arba šalinti.

## 4. Eksperimentinis tyrimas

Mobilių įrenginių rinkoje siūlo daugybę įvairių įrenginių. Skirtingi gamintojai naudoja skirtingas technologijas, programinę įrangą, taip pat vartotojai naudoja skirtingus įrenginio nustatymus. Dėl šių priežasčių taikomoji aplikacija gali nepilnai veikti skirtinguose mobiliuose įrenginiuose. Atlikdamas šį tyrimą sieksiu išsiaiškinti kokiuose mobiliuose įrenginiuose taikomoji aplikacija funkcionuoja nepilnai, kokios to priežastys bei kaip ištaisyti esamus trūkumus.

Eksperimentinis tyrimas bus atliekamas naudojantis trejų skirtingų gamintojų mobiliais įrenginiais: Nokia E65, Nokia 6101, Sony–Ericsson P1 ir Samsung X500. Šie mobilūs įrenginiai turi skirtingo dydžio ekranus bei įdiegta skirtinga programinė įrangą.

**Lentelė 1. Taikomosios aplikacijos bandymas.**

	<b>Nokia E65</b>	<b>Nokia 6101</b>	<b>Sony-Ericsson P1</b>	<b>Samsung X500</b>
<b>Aplikacijos įdiegimas</b>	Sujungus telefoną su kompiuteriu per USB jungtį.	Parsisiunčiant iš interneto arba kito įrenginio per IR sąsają.	Sujungus telefoną su kompiuteriu per USB jungtį.	Atsiunčiant iš kito įrenginio per bluetooth sąsają.
<b>Datos įvedimas</b>	Datos įvedimas iš klaviatūros, tačiau numatytas pasirinkimas iš kalendoriaus.	Datos įvedimas iš klaviatūros, tačiau numatytas pasirinkimas iš kalendoriaus.	Veikia kaip numatyta	Datos įvedimas iš klaviatūros, tačiau numatytas pasirinkimas iš kalendoriaus.
<b>Naujas įrašas</b>	Naujai įvestas įrašas pasislepia po telefone nustatyto fono, įrašas matomas iš naujo įjungus aplikaciją. Atnaujinta aplikacija, t.y. lentelės generavimo vieta.	Veikia kaip numatyta.	Naujai įvestas įrašas pasislepia po telefone nustatyto fono, įrašas matomas iš naujo įjungus aplikaciją. Atnaujinta aplikacija, t.y. lentelės generavimo vieta.	Naujai įvestas įrašas pasislepia po telefone nustatyto fono, įrašas matomas iš naujo įjungus aplikaciją. Atnaujinta aplikacija, t.y. lentelės generavimo vieta.
<b>Įrašo šalinimas ir redagavimas</b>	Veikia kaip numatyta	Veikia kaip numatyta	Veikia kaip numatyta	Veikia kaip numatyta
<b>Padalinio, kliento, instruktoriaus ir automobilio pasirinkimas</b>	Veikia kaip numatyta.	Pasirinkimas įvedamas iš naujai atsidariusio lango, tačiau tai nepatogumų nesukelia.	Veikia kaip numatyta	Veikia kaip numatyta
<b>Duomenų atvaizdavimas</b>	Ekrane telpa 4 lentelės stulpeliai iš 5, todėl vartotojas pakankamai patogiai naudojasi šia aplikacija.	Ekrane telpa tik 2 lentelės stulpeliai iš 5, tai vartotojui sukelia nepatogumų peržiūrint duomenis. Telefonas turi tik 128x160 dydžio rezoliucija, todėl yra nepajėgus atvaizduoti daugiau duomenų viename ekrane.	Telefono ekrane pilnai atvaizduojama visa duomenų lentelė. Taip pat lietimui jautrus ekranas palengvina aplikacijos valdymą.	Ekrane telpa tik 2 lentelės stulpeliai iš 5, tai vartotojui sukelia nepatogumų peržiūrint duomenis. Telefonas turi tik 128x160 dydžio rezoliucija, todėl yra nepajėgus atvaizduoti daugiau duomenų viename ekrane.

Kaip matome iš atlikto eksperimentinio tyrimo, kad ne visos taikomosios aplikacijos funkcijos veikia kaip numatyta, be to įrenginiai jas atlieka skirtingai.

- Visuose telefonuose pavyko įdiegti taikomąją aplikaciją, tačiau Nokia 6101 ir Samsung X500 neturi sujungimo su kompiuteriu galimybes, todėl tai buvo atlikta kitu būdu (žr. 1 lentelė, 1 eilutė).
- Nokia E65, Nokia 6101 ir Samsung X500 mobilūs telefonai neleidžia pasirinkti datos iš kalendoriaus. Datą reikia įvesti klaviatūra. Šios problemos išspręsti nepavyko, nes testuoti Nokia telefonai savyje neturi tokios funkcijos (žr. 1 lentelė, 2 eilutė).
- Įvedant naują įrašą Nokia E65, Sony-Ericsson P1 ir Samsung X500 telefonuose jis pasislepia po nustatyto fonu. Tai svarbiausiai problema atrasta atliekant šį eksperimentinį tyrimą. Ją pavyko išspręsti atnaujinus taikomosios aplikacijos programinį kodą (žr. 1 lentelė, 3 eilutė).



- Įrašo šalinimo ir redagavimo funkcija be priekaištų veikė visuose bandytuose mobiliuose įrenginiuose (žr. 1 lentelė, 4 eilutė).
- Duomenų pasirinkimas iš sąrašo veikė ne taip kaip numatyta Nokia 6101 mobiliame telefone. Šiame įrenginyje duomenys pasirenkami iš naujame lange atsidarančio sąrašo, tačiau tai nepatogumų neturėtų sukelti. Neatitikimo išspręsti nepavyko, nes mobilus įrenginys nepalaiko iššokančio sąrašo galimybės (žr. 1 lentelė, 5 eilutė).
- Pilną duomenų lentelę atvaizduoja tik Sony-Ericsson P1 mobilus telefonas, nes jis turi 2,6“ ekraną su 240x320 rezoliucija. Taip pat lietimui jautrus ekranas supaprastina navigaciją duomenų lentelėje. Nokia 6101 ir Samsung X500 turi mažiausius ekranus iš bandytų mobilių įrenginių. Šie modelis atvaizduoja tik 2 lentelės stulpelius iš 5, o tai labai nepatogu vartotojui (žr. 1 lentelė, 6 eilutė).

Universalios sąsajos taikomosios aplikacijos kūrimas yra ekonomiškai brangus procesas, kadangi aplikacija turi būti testuojama įvairiuose mobiliuose įrenginiuose, kurie sukurti naudojant skirtingas technologijas. Be to daugeliu atveju taikomoji aplikacija neveikia vienodai visuose mobiliuose įrenginiuose. Vystantis mobilioms technologijoms ir aplikacijas kuriant naujiems įrenginiams, senieji nesugeba atlikti funkcijų taip kaip numatyta, todėl jos yra atliekamos supaprastintu būdu. Kaip matėme iš eksperimentinio tyrimo Nokia 6101 telefone įrašai pasirenkami iš naujo lango, tuo tarpu numatytas pasirinkimas yra iš iššokančio sąrašo. Tokio tipo problemas galima išspręsti tik adaptuojant aplikaciją pagal senesnę mobilų įrenginį, tačiau tobulėjant technologijoms tai būtų neracionalus žingsnis.

Išanalizavus šiuos mobilius įrenginius galima teigi, kad geriausiai taikomosios aplikacijos veikimą atitinka Sony-Ericsson P1, o atlikus aplikacijos atnaujinimus šis mobilus įrenginys pilnai atlieka visas aplikacijos funkcijas.

## 5. Išvados

Atlikta mobilių įrenginių programavimo priemonių NetBeans ir Eclipse pagrindinių funkcijų apžvalga bei tarpusavio palyginimas. Nustatyta, kad abu programavimo paketai turi tam tikrų pranašumų, tačiau tolesniam tyrimo plėtojimui pasirinktas NetBeans programavimo įrankis. Šiuo įrankiu sukurta taikomoji aplikacija skirta mobiliems įrenginiams. Aplikacija buvo išbandyta keturiuose mobiliuose telefonuose bei nustatytos blogai veikiančios aplikacijos funkcijos. Atliekant tyrimą aplikacija geriausiai veikė Sony-Ericson P1 mobiliame telefone. Taip pat galime daryti išvadą, kad taikomosios aplikacijos veikimas nepriklauso tik pačios aplikacijos. Didelės įtakos turi techninė ir programinė įranga naudojama mobiliame įrenginyje.

## Literatūra

1. Java Micro Edition (ME) [Paskutinį kartą žiūrėta 2009.03.01]. Prieiga Internetu: <<http://java.sun.com/javame/index.jsp>>
2. NetBeans Mobility features [Paskutinį kartą žiūrėta 2009.03.01]. Prieiga Internetu: <<http://www.netbeans.org/features/javame/index.html>>
3. Eclipse MTJ features [Paskutinį kartą žiūrėta 2009.03.08]. Prieiga Internetu: <<http://www.eclipse.org/home/categories/index.php?category=mobile>>
4. Fregmentation of mobile application [Paskutinį kartą žiūrėta 2009.03.01]. Prieiga Internetu: <<http://www.comp.nus.edu.sg/~damithch/df/device-fragmentation.htm>>