

ŠIAULIŲ UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

EGLĖ BUKIENĖ

Informatikos specialybės II magistratūros kurso
neakivaizdinio skyriaus studentė

**MOBILAUS ELEKTRONINIO DIENYNO KŪRIMO
TECHNOLOGIJOS**

BAIGIAMASIS DARBAS

Darbo vadovas:

prof. habil. dr. G.Kulvietis

Darbo recenzentė:

doc. dr. S.Turskienė

Šiauliai, 2005/2006 m.m.

Turinys:

1 Įvadas.....	3
1.1 Temos aktualumas.....	3
1.2 Mokslinis naujumas.....	3
1.3 Praktinė vertė.....	4
1.4 Tyrimo metodai.....	4
1.5 Darbo tikslas ir uždaviniai.....	4
2 Mobilųjų technologijų analizė.....	5
2.1 Įvadas.....	5
2.2 GSM technologijų analizė.....	5
2.3 Bevielio tinklo technologijų analizė.....	8
2.4 Mobilųjų priemonių architektūra.....	9
2.5 Išvados.....	10
3 Java technologijos.....	11
3.1 Įvadas.....	11
3.2 Java kalbos privalumai ir trūkumai.....	11
3.3 Java platformos.....	13
3.4 WebSphere priemonių šeima.....	14
3.5 Duomenų bazių sinchronizavimo priemonės.....	20
3.6 Išvados.....	24
4 Programų kūrimo priemonių analizė.....	25
4.1 Įvadas.....	25
4.2 Programų kūrimo priemonių analizė.....	25
4.3 Išvados.....	30
5 Mobilaus el. dienyno modelio realizavimas.....	31
5.1 Įvadas.....	31
5.2 Mobilaus el. dienyno specifikacijos.....	31
5.3 Mobilaus el. dienyno realizavimas.....	37
5.4 Išvados.....	44
6 Išvados.....	45
7 Literatūra ir informacijos šaltiniai.....	46
8 Anotacija.....	49
9 Summary.....	50
10 Priedai.....	51
10.1 1 priedas.....	51
10.2 2 priedas.....	59
10.3 3 priedas.....	74

1 Įvadas

1.1 Temos aktualumas

Dar ne taip seniai buvusios naujiena Lietuvoje, mobiliosios technologijos vis sparčiau veržiasi į mūsų kasdienį gyvenimą. Buvęs prabanga, mobilusis telefonas šiandien naudojamas ne tik skambinimui bei trumpųjų žinučių siuntimui, bet ir įvairioms pramogoms: piešinukų, atvirukų, žaidimų, įvairių programų siuntimui, naršymui internete ir kt. Dar daugiau – mobiliuoju telefonu jau galima susimokėti už mašinos parkavimą, rezervuoti viešbutį, užsisakyti prekes, ieškoti kelio vairuojant, atlikti bankines operacijas, valdyti savo būsto informaciją (vandentiekio, šilumos, elektros, vėdinimo ir kt. sistemas), fotografuoti, filmuoti, organizuoti teleposėdžius, išvykus iš biuro prisijungti prie vidinio įmonės kompiuterių tinklo, stebėti ir valdyti kitus mobiliuosius objektus (pvz.: transporto priemones) ir kt.

Bevielių tinklų bei įrenginių technologijos ir tiems įrenginiams pritaikytų paslaugų vystymąsi, ypač pastaraisiais metais, nulėmė tai, jog mobiliųjų technologijų panaudojimas labai išaugo visame pasaulyje.

Taigi, mobiliosios technologijos tampa neatsiejama buities, pramogų, verslo, mokslo, politikos dalimi ir Lietuvoje.

Šio baigiamojo darbo tema – „Mobilusis elektroninis dienyno kūrimo technologijos“. Pati elektroninio dienyno idėja nebėra nauja nei Lietuvoje, nei juo labiau pasaulyje. Vis daugiau šalies mokyklų šalia įprasto dienyno savo iniciatyva sėkmingai naudoja elektroninį dienyną ugdymo procese, be to, planuojama sukurti tokią dienyną, vieningą visoms Lietuvos bendrojo lavinimo mokykloms. Šiame darbe analizuojama galimybė patobulinti iki šiol naudotą el. dienyno formą – dienyno kūrimui panaudojant mobiliąsias technologijas, taip padarant jį pasiekiamą ir valdomą ir mobiliųjų įrenginių pagalba.

1.2 Mokslinis naujumas

Taikomųjų programų lietuvių kalba mobiliesiems įrenginiams vis dar nėra daug – rinkoje vyrauja pramogoms skirti produktai. Ir nors jau yra bandymų sukurti el. dienyną, pritaikytą ir mobiliems vartotojams, jie apsiriboja tik duomenų skaitymu naudojant mobiliojo telefono WAP prieigą ar SMS paslaugas. Šiame darbe pasiūlytas mobilusis el. dienyno modelis leidžia ne tik peržiūrėti, bet ir įrašyti bei redaguoti duomenis, be to yra sukurtas Java pagrindu.

1.3 Praktinė vertė

Mobiliosios technologijos tampa vis svarbesnės, todėl atsiranda vis daugiau siūlymų bei sprendimų, kaip kuo efektyviau panaudoti įvairius mobilius įrenginius ir kaip juos padaryti patogiais ir naudingais darbo įrankiais. Mobilus el. dienynas išplėstų bei papildytų įprastas naudojimosi el. dienynais galimybes, be to, darbe išanalizuotos tokio dienyno kūrimo technologijos būtų naudingos kuriant kitas panašaus pobūdžio programas.

1.4 Tyrimo metodai

Literatūros analizė. Mobiliųjų technologijų įvairovės, techninių galimybių, programavimo galimybių analizė. Platformos, tinkamos mobilaus el. dienyno realizavimui, ir jai skirtų programinių produktų analizė. Programos tobulinimo ir panaudojimo perspektyvų įvertinimas.

1.5 Darbo tikslas ir uždaviniai

Darbo tikslas – išanalizuoti mobilaus el. dienyno kūrimo technologijas. Siekiant darbo tikslo, reikia atlikti šiuos uždavinius:

- išanalizuoti mobiliųjų technologijų įvairovę ir technines galimybes;
- parinkti optimalią platformą programos realizavimui ir išanalizuoti šiai platformai skirtus programinius produktus;
- ištirti programavimo mobiliems įrenginiams priemonių galimybes;
- sukurti veikiantį mobilaus el. dienyno modelį bei numatyti mobilaus el. dienyno tobulinimo ir panaudojimo perspektyvas.

2 Mobilųjų technologijų analizė

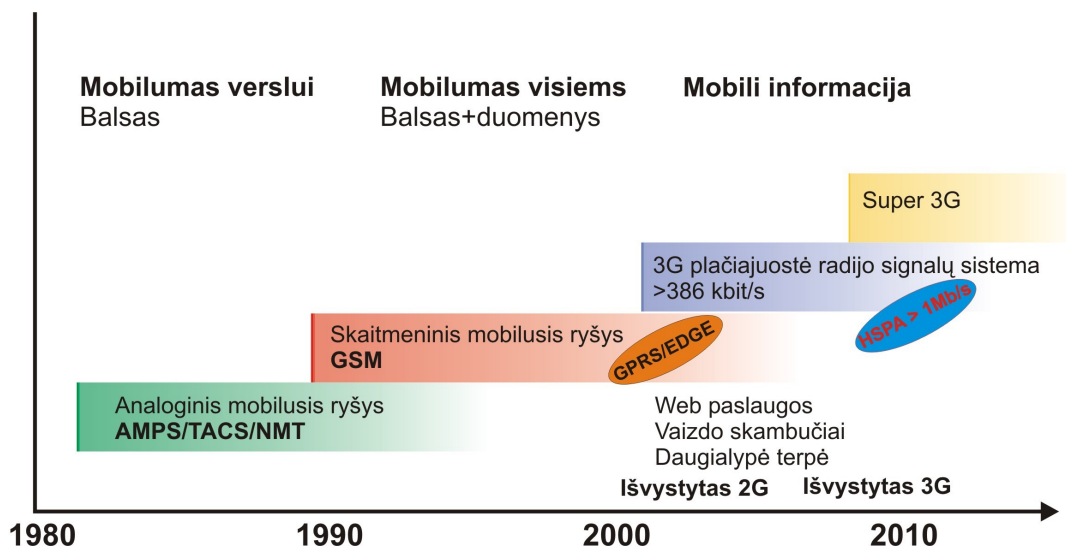
2.1 Įvadas

Nors nuo pirmojo mobiliojo telefono atsiradimo praėjo tik trys dešimtmečiai¹, tačiau mobiliųjų technologijų srityje įvyko ženklūs pokyčiai: nuo analoginio ryšio pereita prie skaitmeninio, nuo balso perdavimo – prie duomenų (garso, teksto, vaizdo) perdavimo.

Dar daugiau – šiuolaikiniai mobilieji įrenginiai savo galimybėmis vis labiau panašėja į personalinius kompiuterius. Dauguma verslo klasės mobiliųjų telefonų, jau nekalbant apie delninius kompiuterius, turinčių atviras operacines sistemas, savo savybėmis (procesoriaus greitaveika, operatyvinės bei nuolatinės atminties kiekiu, grafinio ekrano skiriamąją gebą) nenusileidžia pirmiesiems personaliniams kompiuteriams, o kai kada juos ir lenkia.

2.2 GSM technologijų analizė

GSM technologijų šeimą sudaro tokios technologijos: GSM, GPRS, EDGE, 3GSM. 2.1 pav. pavaizduota GSM technologijų evoliucija: [4]



2.1 pav. GSM technologijų evoliucija

¹ Pirmąjį mobilųjį telefoną 1973 m. sukūrė Martin Cooper. [14]

GSM (Global System for Mobile communications) – globalus mobiliųjų telefonų ryšio standartas, skirtas perduoti duomenims ir balsui. Šia technologija paremta didžioji dalis pasaulio mobiliųjų telefonų tinklų. Nuo pirmos kartos bevielių sistemų GSM skiriasi tuo, jog naudoja skaitmenines technologijas ir laiko padalinimo daugybinės prieigos perdavimo metodus. GSM palaiko duomenų perdavimo greitį iki 9,6 Kbps, kas leidžia teikti paprasčiausias duomenų perdavimo paslaugas, tokias kaip pvz. SMS (Short Message Service). Kita svarbi GSM galimybė – teikti mobilaus ryšio paslaugas ir kito operatoriaus tinkle (roaming), leidžianti vartotojams keliaujant naudotis tomis pačiomis paslaugomis, kaip ir namie. [1]

GPRS (General Packet Radio Service) – GSM mobiliojo ryšio technologija, skirta paketiniam duomenų perdavimui GSM tinkluose. Ši technologija naudoja tuos pačius resursus kaip ir balso telefonija, tačiau naudojamas visiškai kitas signalo perdavimo būdas. GPRS palaiko IP (Internet Protocol) ir PPP (Point-to-Point Protocol) protokolus, kurių pagalba mobilaus telefono vartotojai gali dirbti internete ar perduoti el. pašto pranešimus. GPRS kartais vadinama „2,5G“ technologija, nes yra tarpinis variantas tarp antros (2G) ir trečios (3G) bevielių sistemų kartos. Apie 40 Kbps duomenų perdavimo sparta leidžia naudotis panašiomis galimybėmis kaip ir ryšys per komutuojamą liniją (dial-up), tačiau vartotojui suteikiama laisvė prie interneto prisijungti iš bet kur. [2]

EDGE (Enhanced Data rates for Global Evolution) – skaitmeninė mobilaus ryšio technologija, nauja GPRS (paketinio duomenų perdavimo) versija, dar vadinama EGPRS (Enhanced GPRS). Pagrindinis EDGE privalumas – itin spartus ryšys. Apie 236 Kbps prisijungimo sparta yra ženkliai didesnė nei GPRS technologijos. EDGE greitaveikos užtenka naršyti internete, skaityti elektroninį pašta, parsisiųsti muzikos ar vaizdo klipus, priimti bei siųsti laiškus su prikabintomis bylomis. Prisijungus EDGE ryšiu, galima skambinti internetu ar bendrauti naudojant vaizdo kamerą. [3]

3GSM – paskutinis GSM šeimos narys. Dar vartojamas bendras terminas 3G (3 Generation), apibūdinantis mobiliųjų komunikacijų sistemas, kurios skirtos įvairių paslaugų teikimui, pradedant balso telefonija ir baigiant daugialypės terpės bei didelio greičio Interneto prieigos paslaugomis. 3G technologija leidžia bendrauti tarpusavyje vaizdo skambučiais, siunčiant ir balsą ir vaizdą, bei perduoti duomenis pokalbių metu. Tradiciniame 3G tinkle duomenys gali būti perduodami iki 384 Kbps sparta, prognozuojama, jog šis greitis sparčiai augs. [4]

Super 3G (S3G) – ilgalaikės 3G standarto WCDMA evoliucijos vizija. Jos tikslas yra pasiekti priėmimo (vartotojo terminale) greitį iki 100 Mbps ir sumažinti gaisties laiką iki 10 ms, taip pat

optimizuoti komunikacijų tinklą IP duomenų srautams bei pagerinti transliacijos paslaugų palaikymą jame. [5]

2.1 lentelėje atlikta lyginamoji GSM technologijų analizė:

2.1 lentelė GSM technologijų lyginamoji analizė

Technologija	Duomenų perdavimo sparta	Privalumai	Trūkumai	Tinkamumas
GSM	9,6 Kbps	technologiją palaiko bet kuris skaitmeninio ryšio mobilusis telefonas	nedidelė duomenų perdavimo sparta, netinka daugialypės terpės paslaugoms	tinka balso telefonijai, SMS paslaugoms; galimybė teikti mobilaus ryšio paslaugas ir kito operatoriaus tinkle (roaming)
GPRS	40 Kbps	paketinis duomenų perdavimas, didesnė perdavimo sparta, mokama tik už perduotų duomenų kiekį	sesijos metu kintantis greitis, nepalaikomas faksogramų perdavimas	tinka jei sesijos vyksta dažnai, o vienu metu perduodamas duomenų masyvas yra nedidelis; norint ieškoti informacijos ar skaityti naujienas internete
EDGE	236 Kbps	paketinis duomenų perdavimas, spartus ryšys	palaikoma tik naujo modelio mobiliuosiuose įrenginiuose	tinka naršyti internete, skaityti elektroninį paštą, parsisiųsti muzikos ar vaizdo klipus, priimti bei siųsti laiškus su prikabintomis bylomis; galima skambinti internetu ar bendrauti naudojant vaizdo kamerą
3GSM	384 Kbps	itin spartus ryšys, vaizdo perdavimas	didelis energijos suvartojimas, reikalingas tankesnis bazinių stočių tinklas, brangūs technologiją palaikantys įrenginiai	tinka norint perduoti duomenis pokalbio metu, stebėti televizijos transliaciją realiu laiku; vaizdo bylų atsisiuntimas ir peržiūra, vaizdo skambučiai

2.3 Bevielio tinklo technologijų analizė

IrDA (Infra red Data Assotiation) – protokolas, leidžiantis susijungti su periferiniais įrenginiais be kabelio infraraudonųjų spindulių pagalba. IrDA leidžia palaikyti ryšį nedideliais atstumais – apie 1m., 30 laipsnių kampu, negali įveikti kliūčių, todėl technologija skirta tik tiesioginio matomumo zonoms, be to veikia principu „taškas taškas“. IrDA naudoja ypač aukšto dažnio radijo bangas, kurios yra žemesnės tik už matomą elektromagnetinio spektro spalvą. Šios sistemos nebrangios, saugios, tačiau apima ribotą diapazoną. Nėra patogios paslankiems vartotojams. [6]

Bluetooth arba **BT** (angl. blue tooth – mėlynas dantis) – asmeninių tinklų (PAN) belaidžio ryšio gamybinė specifikacija (pirmąkart pasiūlyta Ericsson įmonės).

Bluetooth leidžia keisti informacija tarp tokių įrenginių kaip delniniai ir paprasti asmeniniai kompiuteriai, mobilieji telefonai, nešiojami kompiuteriai, spausdintuvai ar skaitmeniniai fotoaparatai. Protokolas naudoja patikimą, visuotinai prieinamą artimo ryšio trumpųjų radijo bangų diapazoną (nelicencijuotos 2.45 GHz dažnio ISM bangos). Siekiant išvengti sąveikos su kitais tą patį dažnį naudojančiais protokolais naudojami 1 Mhz pločio 79 kanalai, keičiami 1600 kartus per sekundę.

Bluetooth leidžia įrenginiams bendrauti iki 100 metrų atstumu vienas nuo kito, netgi jei jie yra skirtingose patalpose. [7]

WiFi (Wireless Fidelity) – bevielio greitaegio kompiuterinio tinklo technologija, leidžianti sukurti infrastruktūrą mobilaus interneto ryšio teikimui. Teritorija, kur yra įdiegta WiFi infrastruktūra (ryšio zona) vadinama hotspotu. HotSpot'e galima naudotis greitaegiu interneto ryšiu bevieliu būdu. HotSpot'ai paremti bevieliu kompiuteriniu IEEE-802.11b (WiFi) standarto ryšiu, siekiant užtikrinti saugų, patikimą bei greitą bevielį ryšį. Šis ryšys buvo sukurtas kaip alternatyva laidiniam ofiso kompiuterių tinklui (LAN), todėl neretai jis dar vadinamas Wireless LAN (WLAN). WiFi tinklas gali būti naudojamas kompiuterių sujungimui vienas su kitu, prijungimui prie interneto ar įprastų vielinių kompiuterinių tinklų (kurie paremti IEEE 802.3 standartu). WiFi bevelis ryšys veikia 2,4 ir 5 GHz nelicencijuojamo dažnio diapazone, o greitis siekia 11Mbps (802.11b standarto) ar 54 Mbps (802.11a standarto). [8]

WiMax (Worldwide Interoperability for Microwave Access) – bevielio duomenų perdavimo standarto 802.16, priimto 2003m. sausį, komercinis pavadinimas. Standartas suprojektuotas specialiai darbui dideliais atstumais. Kitaip nei WiFi, WiMax mažiau susietas su konkrečiu diapazonu – jo variantai pritaikyti dažniams nuo 2 iki 11GHz ir nuo 10 iki 66 GHz. Vis tik didžiausias WiMax privalumas – galimybė dirbti dideliais atstumais. Atstumas tarp įrenginių gali siekti 50 km, be to, tarp jų gali nebūti

tiesioginio matomumo. Signalo galia ir didelis atsparumas atspindžiams įgalina WiMax veikti ten, kur negali WiFi.

WiMax tinklai apima plačią duomenų perdavimo sritį: nuo paprasto greitaeigio namų vartotojų prisijungimo iki internetinės telefonijos, verslo, valstybinių ir kitų organizacijų ryšių poreikių. [9]

2.2 lentelėje atlikta bevielio tinklo technologijų lyginamoji analizė:

2.2 lentelė Bevielio tinklo technologijų lyginamoji analizė

Technologija	Standartas	Dažnis	Atstumas	Duomenų perdavimo sparta	Privalumai	Trūkumai
IrDA	IrDA-standart	Infraraudonieji spind.	1 m	16 Mbps	didelė sparta, žema kaina	„taškas į tašką“ sujungimas, reikalingas tiesioginis matomumas
Bluetooth	IEEE 802.15	2,45 GHz	100 m	3 Mbps	saugumas	mažas atstumas, nedidelė duomenų perdavimo sparta
WiFi	IEEE 802.11	2,4 GHz ir 5GHz	90 m	54 Mbps	didelė perdavimo sparta, saugumas	sąlyginai mažas atstumas, didelis energijos suvartojimas
WiMax	IEEE 802.16	2-11 GHz, 10-66 GHz	50 km	70 Mbps	didelė perdavimo sparta, atsparumas atspindžiams, darbas dideliais atstumais	didelis energijos suvartojimas

2.4 Mobilųjų priemonių architektūra

Atsižvelgiant į mobiliųjų ryšio priemonių architektūrą, galima būtų išskirti tris pagrindines jų grupes: mobilieji telefonai, „protingieji“ telefonai (smartphone) ir delniniai kompiuteriai (PDA – Personal Digital Assistant). Vis sparčiau vystantis mobiliosioms technologijoms skirtumai tarp šių įrenginių tipų nyksta, todėl neretai kyla problemų dėl nevienodo sąvokų interpretavimo. Pagrindinė ypatybė, skirianti „protingąjį“ telefoną nuo įprasto – pakankamai išvystyta operacinė sistema. Tai leidžia

įdiegti papildomas programas, sukurtas ne tik telefono gamintojo, bet ir kitų programinės įrangos kūrėjų. Nuo delninių kompiuterių, turinčių GSM ryšio galimybę (kartais vadinamų komunikatoriais), „protingieji“ telefonai skiriasi šiek mažesniu dydžiu ir yra labiau pritaikyti tiesioginei paskirčiai – šnekėti. Paprastai „protingieji“ telefonai turi ir tradicinę telefoninę klaviatūrą. 2.3 lentelėje palygintos mobiliųjų telefonų, „protingųjų“ telefonų ir delninių kompiuterių architektūros: [10, 11, 12]

2.3 lentelė Mobiliųjų priemonių architektūros lyginamoji analizė

	Mobilieji telefonai	„Protingieji“ telefonai	Delniniai kompiuteriai
Ekranas	~170x220 taškų, 65000 spalvų.	240x320 taškų, 65000 spalvų, jautrus lietimui	640x480 taškų, 65000 spalvų, jautrus lietimui
Procesoriaus dažnis	Nėra duomenų	iki 300 MHz	~500 MHz
Vidinė atmintinė	~40MB	~100 MB	Nuo 32 MB iki 128 MB
Sąsaja su kompiuteriu	IrDA, Bluetooth	IrDA, Bluetooth, WiFi	IrDA, Bluetooth, USB, WiFi
Operacinė sistema	–	Symbian OS, Palm OS, Windows Mobile, Pocket PC	Pocket PC, Palm OS, Symbian, MS Smartphone, Linux

2.5 Išvados

Bevielio duomenų perdavimo greičiui artėjant prie fiksuoto interneto vartotojams teikiamų greičių ir vis tobulėjant mobiliųjų įrenginių galimybėms, pastarieji vis dažniau pakeis asmeninį kompiuterį: tampa paprasta į mobiliųjų įrenginių atsisiųsti ne tik tekstinę, bet ir vaizdinę informaciją, paslaugų spektras išplito pradedant balso telefonija ir baigiant daugialypės terpės bei didelio greičio interneto prieigos paslaugomis.

Nors bevielio tinklo technologijomis perduodamų duomenų greitis nepranoksta įprastų vielinių tinklų, pastarųjų bus vis dažniau atsisakoma dėl bevielinių tinklų eksploatavimo patogumo.

Vystantis technologijoms mažėja skirtumas tarp mobilaus telefono ir delninio kompiuterio – mobilus telefonas įgauna vis daugiau funkcijų, anksčiau būdingų tik delninukams.

3 Java technologijos

3.1 Įvadas

Java – Sun Microsystems kompanijos sukurta objektiškai orientuota programavimo kalba, oficialiai pristatyta 1995 metais. Ji skirta tiek savarankiškų autonominių programų, tiek tinklu persiunčiamų ir interneto naršyklių vykdomų klientinių programų (applets), tiek serverinių programų (servlets) kūrimui.. Lyginant su kitomis programavimo kalbomis, pagrindinis Java pranašumas yra jos nepriklausomumas nuo kompiuterių tipo. [33]

Java laikoma revoliuciniu programavimo žingsniu pirmiausia dėl programavimo galimybių internete – kalbant tiek apie klientinį, tiek apie serverinį programavimą. Pagrindinė klientinių-serverinių sistemų idėja ta, jog centralizuotas informacijos bankas yra serveryje, į kurį gali kreiptis grupė klientinių kompiuterių ar mobiliųjų įrenginių. Serveryje saugomos informacijos pokyčiai greitai prieinami klientiniams įrenginiams.[33]

Dauguma kompanijų integruoja Java palaikymą į savo serveriams skirtą programinę įrangą, ne išimtis ir IBM, tapusi viena lyderių Java technologijų panaudojime – Java pasirinkusi galingos aplikacijų kūrimo platformos WebSphere pagrindu.

3.2 Java kalbos privalumai ir trūkumai

Privalumai: [15, 37]

Paprasta. Java buvo sukurta remiantis populiariausios objektiškai orientuotos kalbos C++ sintakse ir Smalltalk bibliotekomis bei pašalinant konstrukcijas, sąlygojančias programavimo klaidas – rodykles, dinaminį atminties valdymą, daugialypį paveldėjimą (multiple inheritance), operatorių perdengimą, parametrizavimo šablonus (templates). Tiesa, Java 2 1.5 šablonai buvo įvesti.

Objektiškai orientuota. Java palaiko visas pagrindines objektiškai orientuotos programavimo kalbos konstrukcijas – klases ir objektus, paveldėjimą, sąsajas ir jų realizavimą, informacijos slėpimą (encapsulation) pagrįsta matomumo (visibility) specifikavimu, metodų perdengimą (overriding) ir perkrovimą (overloading), polimorfizmą, skirstymą į paketus.

Orientuota tinklinių sistemų kūrimui. Java turi standartines bibliotekas tinklinių programų kūrimui bei paprastam paskirstytų objektų realizavimui, taip pat specialias programas – appletus – vykdymui Internet naršyklėse.

Atspari klaidoms. Dėl paprastumo, saugumo bei išimčių valdymo (exception handling) ir neseniai įdiegto sąlygų patikrinimo (assertions) mechanizmo Java yra atspari klaidoms ir leidžia kurti aukštos kokybės programinę įrangą.

Saugi. Java kalboje yra įdiegtas saugumo mechanizmas, kuris patikrina ar programinis kodas nėra kenksmingas (malicious), kiekvienam veiksmui, kuris keičia programos išorę, pvz. sukuria, šalina arba modifikuoja failus, reikalinga tiesiogiai suteikti programai atitinkamus leidimus (permissions). Be to Java kalba neturi konstrukcijų, leidžiančių tiesiogiai valdyti atmintį, atlikti žemo lygio operacinės sistemos (OS) funkcijas – visos Java programos bendrauja su OS per JVM.

Nepriklausoma nuo architektūros. Java programavimo kalboje visi kintamųjų tipai yra griežtai apibrėžti, todėl nekyla tokių problemų kaip kintamas sveikųjų skaičių reikšmių diapazonas priklausomai nuo architektūros – architektūrų skirtumus išsprendžia JVM.

Nepriklausoma nuo operacinės sistemos. Java programos yra sukompilijuojamos į tarpinį kodą (bytecode), kuris yra vykdomas Java Virtualios Mašinos (JVM) interpretatoriaus. Kadangi JVM programinė įranga yra realizuota visoms pagrindinėms OS, teoriškai Java programos gali būti pernešamos ir vykdomos sukompilijuotos tarp įvairių OS. Praktikoje visgi atsiranda problemų dėl skirtingų JVM realizacijų, jų defektų arba abstrakčios vartotojo sąsajos nesuderinimo prie OS specifikos.

Interpretuojama. Java kalba yra interpretuojama ir tai lemia jos nepriklausomumą nuo architektūros ir OS, saugumą bei paprastumą, kadangi interpretavimo aplinkoje gali būti atliekami įvairūs pagalbinių servisai, pvz. automatinis atminties laisvinimas (automatic garbage collector).

Dinamiška. Java programavimo kalba labai sparčiai vystosi, todėl labai dažnai atsiranda įvairių naujų savybių, bibliotekų, technologijų, leidžiančių efektyviau kurti programinę įrangą.

Daugiasrautė. Java leidžia rašyti programas, vienu metu vykdančias kelis srautus toje pat programos adresinėje erdvėje. Yra srautų sinchronizavimo ir tarpšrautinio bendravimo galimybės.

Atvira. Java nuo pat pradžių buvo kuriama kaip nemokama sistema – visos jos standartinės bibliotekos ir pagrindiniai kūrimo įrankiai – kompiliatorius, interpretatorius, dokumentacijos generatorius ir kt. – yra nemokami ir laisvai prieinami. Galima prieiti prie standartinių bibliotekų išeities kodo, viešai registruoti surastus defektus, dalyvauti kuriant naujas Java technologijas. Šiuo metu Java turi bene labiausiai išvystytą technologinę platformą paskirstytų kliento/serverio sistemų kūrimui.

Trūkumai: [15, 37]

Nepakankamas greitis. Kadangi Java yra interpretuojama, Java programų veikimas daugeliu atveju yra lėtesnis už sukompilijuotas programas parašytas C/C++, Fortran, Pascal arba kitomis kompilijuojamomis programavimo kalbomis. Tačiau šita problema dažniausiai iškyla tik kuriant atskiras programas (desktop applications) su grafine vartotojo sąsaja – testai (benchmarks) rodo, kad kliento/serverio programų veikimas yra dažnai net greitesnis už analogiškas C++ realizacijas.

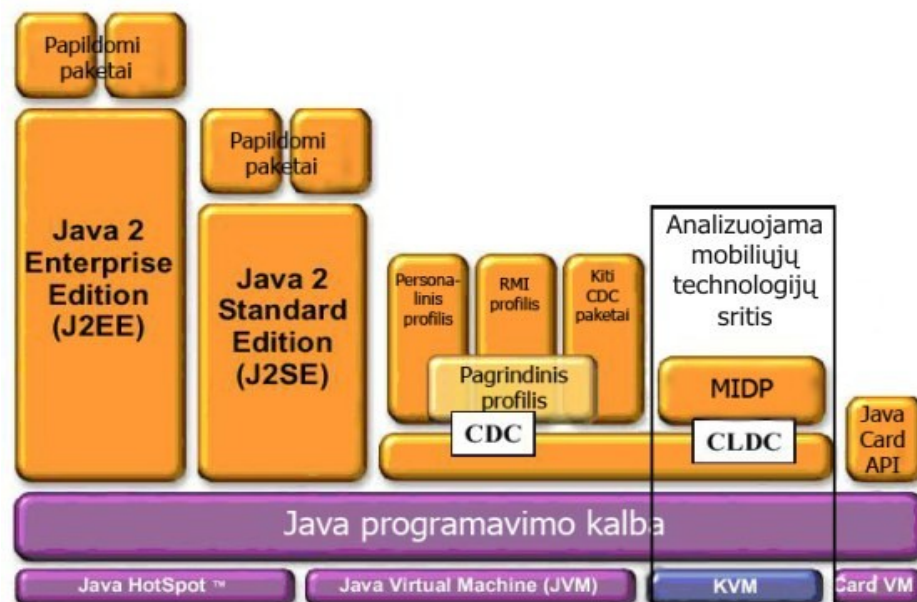
Priklausomybė nuo JVM. JVM jau minėjome kaip privalumą, tačiau tai dažnai yra ir trūkumas kadangi: 1) galutinis vartotojas būtinai turi susiinstaliuoti JVM; 2) galimi skirtingoms Java versijoms parašytų programų ir kompiuteryje suinstaliuotų skirtingų JVM versijų konfliktai.

Primityvūs duomenų tipai. Primityvių duomenų tipų kintamieji Java kalboje yra traktuojami kitaip nei objektai – jais manipuluojama kaip reikšmėmis, o ne kaip nuorodomis (reference). Šis skirtumas apsunkina kalbos mokymą bei įneša galimybių padaryti programavimo klaidų.

Konstruktijų trūkumas. Vardan paprastumo, Java nepalaiko kai kurių konstrukcijų, kurios programuotojams gali būti reikalingos, pvz. objekto būsenos nekintamumo suvaržymo, kuris C++ kalboje buvo atliekamas naudojant raktinį žodį const. Tik naujausioje Java 2 1.5 versijoje buvo įvesti parametrizavimo šablonai.

3.3 Java platformos

Šiuo metu išskiriamos keturios specializuotos Java technologijų platformos. Visa Java 2 platforma pavaizduota 3.1 pav.: [16]



3.2 pav. Java 2 platforma

Java 2 Platform Standard Edition (J2SE) – platforma, apimanti pagrindinius Java kūrimo įrankius bei svarbiausias Java kalbos bibliotekas, kurios yra naudojamos ir kitų platformų, bei atskirų programų (desktop application) kūrimui skirtas bibliotekas, tokias kaip grafinės vartotojo sąsajos komponentų biblioteka Swing. J2SE platformos technologijos yra skirtos atskirų programų, nesudėtingų

kliento/serverio programų ir appletų kūrimui. Net ir ši paprasčiausia dalis jau visiškai palaiko standartų sistemą koordinuotam kelių programų darbui internete (CORBA). [17, 37]

Java 2 Platform Enterprise Edition (J2EE) – platforma, apibrėžianti technologijas skirtas komponentiniam daugiasluoksnių (multi-tier) industrinių (enterprise) programų kūrimui. J2EE platforma yra pagrįsta J2SE platforma ir apibrėžia papildomus servisus, įrankius ir API (Application Programming Interface), palengvinančius industrinių programų kūrimą. Šios technologijos pagal specifikacijas yra realizuojamos J2EE serveriuose, kurie naudojami J2EE programų kūrimui ir įdiegimui. [18, 37]

Java 2 Platform Micro Edition (J2ME) – rinkinys technologijų ir specifikacijų skirtų programuoti buitiniams ir techniniams įrenginiams: mobiliems telefonams, delniniams kompiuteriams (PDA), spausdinimo, TV ir kitiems įrenginiams. Apie šią platformą plačiau kalbėsime 4 skyriuje. [19, 37]

Java Card – technologija, kuri adaptuoja Java platformą kurti programas „protingoms“ kortelėms (smart card) ir kitiems panašioms įrenginiams su labai ribotais atminties ir skaičiavimų resursais. [20, 37]

3.4 WebSphere priemonių šeima

IBM WebSphere – programinės įrangos produktų šeima, skirta aplikacijų infrastruktūrai, verslo sistemų integravimui, portalams, elektroninei komercijai, mobiliajai prieigai ir prieigai balsu. Tai nauja strateginė IBM programų platforma, pagrįsta atvirais standartais ir Java (J2EE) technologijomis.

3.4.1 WebSphere Application Server

WebSphere Application Server – Websphere šeimos pagrindas, kurį sudaro serverių, realizuojančių J2EE specifikaciją, rinkinys. Tai reiškia, kad bet kokia tinklo aplikacija, parašyta laikantis J2EE specifikacijos, gali būti įdiegta bet kokiame WebSphere Application Server serveryje. Toks serveris pateikia programų, sukurtų J2EE ir webservisų pagrindu, vykdymo terpę ir valdymo instrumentus. WebSphere Application Server serveriai pateikiami įvairių, specifinius poreikius atitinkančių konfigūracijų - taip užtikrinama plati aplikacijų diegimo aplinka, atsiranda didesnės galimybės valdyti transakcijas bei saugumą, našumą, prieinamumą, prisijungimą ir kitų WebSphere šeimos produktų pritaikymą.

WebSphere Application Server serveriai palaiko platų platformų spektrą, tame tarpe Unix platformas, Microsoft, IBM z/OS, iSeries operacines sistemas.

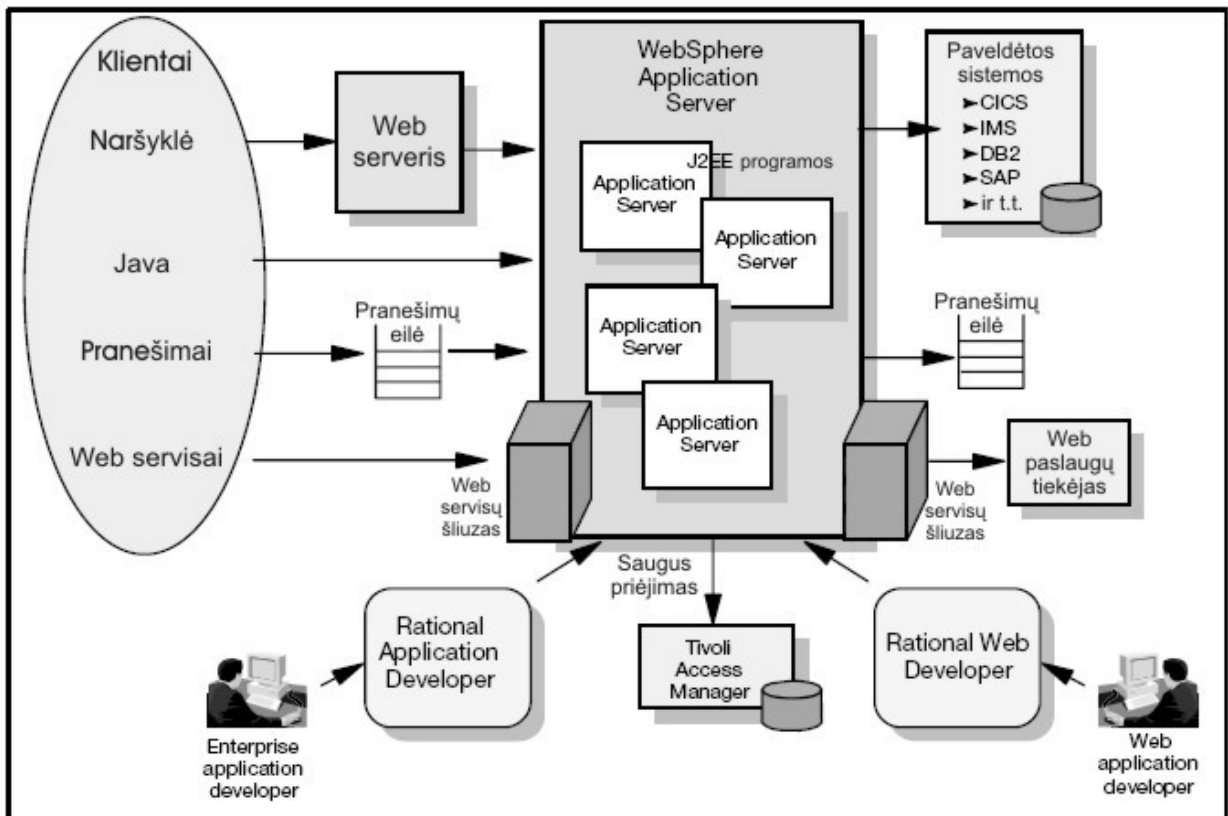
WebSphere Application Server charakteristikos: [21]

- visiškas J2EE V1.4 suderinamumas, o taip pat webservisų palaikymas, viršijant specifikacijas;
- naujas paketavimo būdas, užtikrinantis visišką J2EE V1.4 suderinamumą – nuo pagrindinio Express lygmens paslaugų teikimo iki sudėtingiausio tinklo diegimo, kartu teikiant sudėtinio

(daugybinio) diegimo pasirinktis nuo pavienio serverio iki daugybės galimų plačios apimties konfigūracijų;

- greitas kūrimas ir diegimas, pagreitinantys kūrimo ciklą ir padidinantys galimybę panaudoti egzistuojančius įgūdžius bei išteklius;
- glaudi integracija su IBM Rational įrankiais, produktyvi projektavimo aplinka, Eclipse (atviro kodo integruota kūrimo aplinka) paremta projektavimo aplinka;
- daugiaplatforminis palaikymas.

3.2 pav. pavaizduota principinė WebSphere Application Server funkcionavimo schema:



3.3 pav. Principinė WebSphere Application Server funkcionavimo schema

WebSphere Application Server programinių produktų šeimą sudaro: [21]

WebSphere Application Server-Express – siūlo prieinamą įėjties tašką į e-verslą – „out-of-the-box“ (paruoštą naudojimui, gatavą) sprendimą nesudėtingų, bet dinamiškos interneto svetainių valdymui kartu su lengvai naudojamu tinklo aplikacijos serveriu ir kūrimo aplinka;

WebSphere Application Server – WebSphere šeimos pagrindas, svarbiausias J2EE ir tinklo servisų aplikacijos serveris, užtikrinantis didelį našumą ir lengvai pritaikomą transakcijų variklį dinamiškoms e-verslo aplikacijoms;

WebSphere Application Server Network Deployment – užtikrina padidinto našumo veikimo aplinką ir prieinamumo galimybes, palaikant dinamiškas aplikacijos aplinkas. Be visų ypatybių ir funkcijų, būdingų WebSphere Application Server, ši konfigūracija teikia sudėtingesnius diegimo servišus, į kuriuos įeina grupavimo, „edge-of-network“ servišai, tinklo servisų tobulinimas ir didelis prieinamumas prie paskirstytų konfigūracijų;

WebSphere Extended Deployment – kartu su WebSphere Application Server Network Deployment, užtikrina didelį našumą, lengvai valdomą ir dinamiškai pritaikomą aplinką paskirstytoms WebSphere aplikacijoms, o tai atspindi įrodytų IBM sistemų principus ir idėjas;

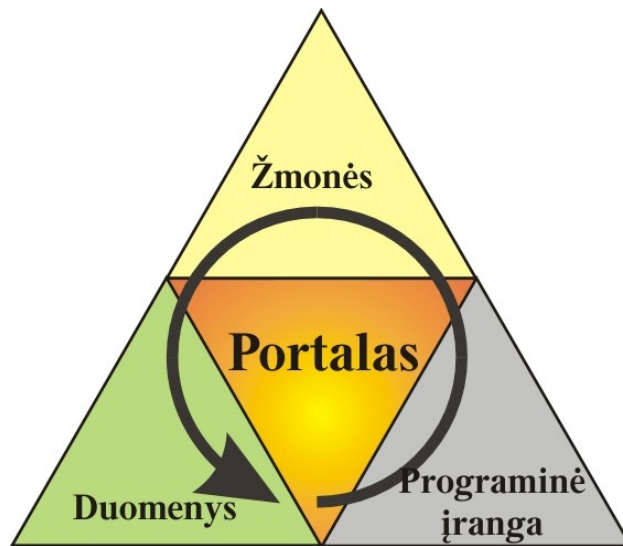
WebSphere Application Server for z/OS – pasižymi tokiu pačiu programavimo modeliu, kaip ir WebSphere Application Server, tačiau konfigūracija optimizuota tam, kad būtų panaudojamos unikalios servisų, kuriuos teikia IBM iSeries™ techninė įranga ir z/OS™ operacinė sistema, savybės;

IBM Rational Web Developer – optimizuota WebSphere programinei įrangai, Eclipse paremta IDE (integruotą kūrimo aplinką) skirta tinklo, tinklo servisų ir Java aplikacijų kūrimui, testavimui ir diegimui;

IBM Rational Application Developer – optimizuota WebSphere programinei įrangai, Eclipse paremta IDE greitam projektavimui, kūrimui, analizei, testavimui, tinklo ir tinklo servisų, J2EE ir Portal aplikacijų diegimui. Apie šį produktą plačiau kalbėsime 4 skyriuje.

3.4.2 WebSphere Portal

IBM WebSphere Portal siūlo vieningą personalizuotą aplinką darbui su įvairiomis aplikacijomis, darbo sekomis, turinio valdymui, bendravimui, taip suteikdama vartotojui vientisumo pojūtį. 3.3 pav. pavaizduota principinė portalo funkcionavimo schema:



3.4 pav. Principinė portalo funkcionavimo schema

WebSphere Portal charakteristikos: [22]

- atvirų standartų (tokių kaip JSR 168) palaikymas;
- greita serverinių aplikacijų integracija (paruoštų portletų ir portletų kūrimo įrankių dėka);
- bendras prieigos taškas;
- vieninga unifikuota sąsaja;
- centralizuotas valdymas;
- aukštas valdymo lygis;
- dokumentų valdymo ir redagavimo įrankiai;
- kolektyvinio darbo įrankiai;
- bendradarbiavimo galimybės (el. žinutės, atskiros darbo vietos darbuotojų komandoms, asmenų paieška, el. susitikimai).

WebSphere Portal programinių produktų šeimą sudaro: [22]

WebSphere Portal for Multiplatforms – jungia programinius produktus ir priemones, skirtas saugių portalų darbui su verslo partneriais (B2B), klientais (B2C) ir bendradarbiais (B2E) kūrimui ir valdymui;

WebSphere Portal-Express for Multiplatforms – padeda mažo ir vidutinio verslo įmonėms lengvai išdėstyti modernius portalus darbuotojams, verslo partneriams ir klientams;

WebSphere Portal Enable for z/OS – portalas verslo įmonėms, reikalaujančioms aptarnavimo greičio ir patikimumo;

WebSphere Portlet Factory – greito WebSphere Portal portletų kūrimo RAD (rapid application development) sistema;

WebSphere Everyplace Mobile Portal Enable – WebSphere Portal galimybių išplėtimas mobiliems įrenginiams;

WebSphere Commerce Portal – portalo pardavimų, pirkimų ir informacijos kanalo valdymo sprendimų papildymas naujausiomis industrinėmis technologijomis.

3.4.3 WebSphere Everyplace

WebSphere Everyplace programų grupė atlieka tas pačias funkcijas, kaip ir IBM Lotus Domino Everyplace, tik yra realizuota WebSphere platformoje. Šių produktų kompleksas leidžia pasiekti informaciją ir kreiptis į serverio programas iš bet kurios vietos, bet kuriuo mobiliuoju ryšio įrenginiu. WebSphere Everyplace pagalba galima organizuoti internetinių puslapių ir serverio programų sąsajų atvaizdavimą mobiliojo įrenginio ekrane, be to, paslaugų kūrėjams ir teikėjams suteikiami įrankiai patogesnių programų kūrimui. Patys mobilieji įrenginiai gali būti valdomi įvairių OS, tame tarpe Palm OS, Linux, Symbian ir Windows Mobile. [23]

WebSphere Everyplace programinių produktų šeimą sudaro: [23]

IBM Workplace Client Technology, Micro Edition – platforma, pritaikyta ateinančios kartos elektroninio verslo aplikacijoms;

WebSphere Everyplace Access – praplečia WebSphere platformą aplikacijoms suteikiant mobiliųjų įrenginių palaikymą, automatizuoja efektyvią iš bet kurios vietos prisijungusio vartotojo sąveiką su duomenimis, aplikacijomis ir verslo procesais;

WebSphere Everyplace Device Manager – skirtas paslaugų tiekėjams palengvinti įvairių bevielių ir mobilių įrenginių valdymą;

WebSphere Everyplace Connection Manager – nuotolinė prieiga prie esamų aplikacijų, užtikrinanti aukštą saugumo lygį ir vientisą tarptinklinį ryšį;

WebSphere Everyplace Service Delivery – platforma, apjungianti ir integruojanti IBM tarpplatforminės programinės įrangos (middleware) infrastruktūrą, reikalingą diegti, valdyti ir derinti tinklo aplikacijas.

3.4.4 WebSphere Transcoding

IBM WebSphere Transcoding Publisher Version 4.0 for Multiplatforms – serveriams skirta programinė įranga, dinamiškai verčianti tinklo turinį ir aplikacijas į įvairias ženklinimo kalbas (multiple

markup language) ir optimizuoja ją pateikimui į mobilius įrenginius, tokius kaip mobilieji telefonai ir delniniai kompiuteriai. Šis sprendimas adaptuoja, pertvarko ir filtruoja turinį, pritaikydamas jį atvaizdavimui populiarijauose įrenginiuose, taip suteikdamas geresnį priėjimą prie mobiliųjų vartotojų – darbuotojų, verslo partnerių, klientų. [24]

Transcoding Publisher leidžia įmonei komunikuoti su savo mobiliais klientais ir darbuotojais, naudojantis informacija, kuri egzistuoja jos informacijos sistemoje. Tam nėra būtinybės kurti ir palaikyti keleto skirtingų versijų ir formato tinklapių. Kadangi perkodavimas yra atliekamas realiu laiku, originalaus turinio pakeitimai yra automatiškai matomi visuose galimuose formatuose.

WebSphere Transcoding charakteristikos: [24]

- vartotojo poreikius atitinkanti personalizacija: informacijos apie vartotoją, esančios centriniame vartotojų kataloge, prieinamumas turinio pritaikymui pagal konkretaus vartotojo pasirinkimą;
- Voice XML perkodavimas: HTML formato vertimas į Voice XML tolimesniam apdorojimui balso naršykle, pvz. WebSphere Voice Server;
- papildomas mašininio vertimo modelis: atpažįsta balsinius parametrus, nustatytus užklausa pateikuso įrenginio naršyklėje, tuomet užklaustas puslapis siunčiamas į WebSphere Translation Server vertimui ir tolimesniam dokumento perkodavimui, po kurio dokumentas gali būti siunčiamas į užklausa pateikusį įrenginį;
- išplėstas kūrimo priemonių rinkinys: šie instrumentai, ypač XSL Stylesheet Editor ir anotacijų redaktorius, padeda greičiau kurti, papildyti ir derinti savo programinius sprendimus, be to jie leidžia padidinti sąnaudų atsiperkamumą pakartotino aktyvų panaudojimo dėka;
- standartinės funkcijos: turinio fragmentacija, keletas lanksčių vystymo variantų, Palm, PocketPC platformų palaikymas.

3.4.5 WebSphere Voice

WebSphere Voice leidžia organizuoti patogų priėjimą prie duomenų iš bet kurios vietos, bet kuriuo metu, naudojantis bet kokia ryšio priemone bei suteikia kalbinės sąsajos panaudojimo galimybę keitimuisi duomenimis. Gausi WebSphere Voice programų šeima turi kalbos atpažinimo ir garsinio atsakymo, pranešimų apdorojimo tarnybų ir vertimo technologijas bei kūrimo priemones daugiau nei dešimčiai kalbų porų. [25]

WebSphere Voice programinių produktų šeimą sudaro: [25]

WebSphere Voice Application Access – tarpplatforminė kalbos programinė įranga, išplečianti WebSphere portalo infrastruktūrą ir programavimo modelį balsui.

WebSphere Voice Server – programinė įranga, skirta dialoginiams e-verslo sprendimams kurti ir diegti.

WebSphere Voice Response for Windows – universali balso apdorojimo platforma, kuri užtikrina išplėstas funkcines galimybes Interactive Voice Recognition (interaktyvaus balso atpažinimo) aplikacijoms Windows operacinėje sistemoje.

WebSphere Voice Response for AIX – universali balso apdorojimo platforma, užtikrinanti išplėstas funkcines galimybes Interactive Voice Recognition (interaktyvaus balso atpažinimo) aplikacijoms, kurios tinka didelėms įmonėms arba telekomunikacijų verslui.

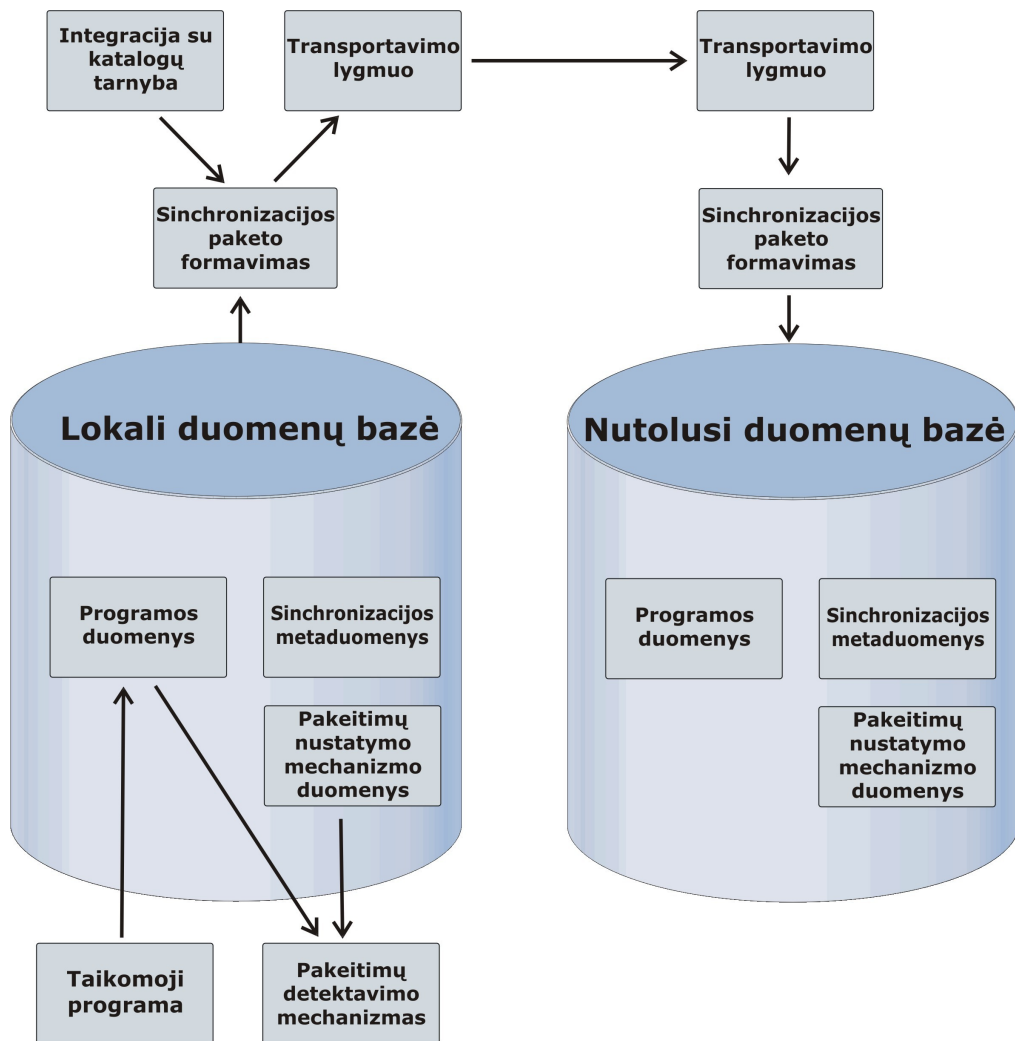
Unified Messaging for WebSphere Voice Response – unifikuotas pranešimų mainų sprendimas, tvarkantis darbuotojo ir kliento balso paštą, el. paštą bei faksus ir užtikrinantis priėjimą per telefoną arba internetą praktiškai bet kur ir bet kuriuo laiku.

3.5 Duomenų bazių sinchronizavimo priemonės

Viena aktualiausių šiandienos informacinių sistemų problemų – duomenų saugojimas. Darbas teritoriškai nutolusiuose filialuose, klientų aptarnavimo padaliniuose, siuntų pristatymo tarnybose ir pan. neįmanomas be tos organizacijos duomenų bazės, kurioje informacija nuolat kinta. Pastarasis faktas įtakoja duomenų sinchronizavimo būtinybę.

Suprasdami problemos aktualumą, daugelis reliacinių duomenų bazių valdymo sistemų (RDBVS) kūrėjų, kartu su savo produktu siūlo ir sprendimus, skirtus duomenų replikavimui. Tačiau šiandieninėje praktikoje ganėtinai dažnai išskyla būtinybė toje pačioje informacinėje sistemoje naudoti skirtingų gamintojų RDBVS, todėl duomenų sinchronizavimo uždavinys tarp skirtingų sistemos dalių tampa ne taip lengvai sprendžiamas. Problemą dar labiau paaštrino bevielių duomenų perdavimo technologijų, tuo pačiu ir mobiliųjų vartotojų (darbuotojų, klientų) atsiradimas.

3.4 pav. pavaizduota principinė duomenų sinchronizavimo schema, atspindinti sinchronizacijos procese dalyvaujančių komponentų sąveiką:



3.5 pav. Principinė duomenų synchronizavimo schema

Vartotojų darbo su duomenimis procese gali iškilti situacija, kuomet vienas ir tas pats įrašas yra modifikuojamas skirtinguose sistemos mazguose. Tokia situacija vadinama konfliktu. Konfliktų sprendimo mechanizmo efektyvumas viena svarbiausių duomenų bazių synchronizavimo priemonių charakteristikų.

Toliau apžvelgiami DB2 ir Lotus kompanijų siūlomi duomenų sinchronizavimui skirti produktus, pritaikyti sinchronizuoti mobilius įrenginius su asmeninio kompiuterio ar organizacijos informacinės sistemos programine įranga.

3.5.1 DB2 Everyplace Sync Server

Naudojant DB2 Everyplace Sync Server (sinchronizavimo serverį) ir DB2 Everyplace Sync Client (sinchronizavimo klientą) galima sinchronizuoti duomenis ir taikomas programas tarp DB2 Everyplace kliento įrenginių ir pagrindinės duomenų bazės. Duomenų sinchronizavimas gali būti dvikryptis arba vienakryptis. Duomenys gali būti atnaujinami DB2 Everyplace mobiliajame įrenginyje arba duomenų bazėje. Pavyzdžiui, vartotojai gali atsisiųsti duomenų poaibį iš DB2 duomenų bazės skirtos z/OS, į DB2 Everyplace duomenų bazę mobiliame įrenginyje, peržiūrėti duomenis, juos keisti ir tada sinchronizuoti pakeistus duomenis į z/OS serverį. Sinchronizacijos serveris taip pat turi konfliktų sprendimo mechanizmą. [26]

DB2 Everyplace Sync Server geba sinchronizuoti reliacinius duomenis su šiomis duomenų bazių valdymo sistemomis: [26]

- DB2 Universal Database V8.1 ir V8.2 Windows, Linux, AIX, HP-UX ir Solaris operacinėms sistemoms;
- DB2 Universal Database V7.1 ir V8.1 z/OS operacinei sistemai;
- DB2 Universal Database V5R1 ir V5R2 IBM iSeries operacinei sistemai;
- Informix Dynamic Server 9.4 (su Informix JDBC tvarkykle);
- IBM Cloudscape V10;
- Lotus Domino Server 6.0.2;
- Oracle 9i and 10g;
- Microsoft SQL Server 2000 su Service Pack 3a (V8.00.760) (su SQL Server JDBC tvarkykle);
- Sybase Adaptive Server Enterprise 11.93 (su JDBC skirtu Sybase J-Connect 5.5).

Pagrindinės funkcijos: [26]

- patobulintos sinchronizacijos funkcijos – konfliktų sprendimas ir duomenų skaidymas;
- duomenų suspaudimas ir šifravimas, užtikrinantis greitesnę, saugią duomenų sinchronizaciją;
- didelis prieinamumas ir apkrovimą balansuojančios konfigūracijos, naudojantis IBM WebSphere Application Server grupėmis; nesudėtinga daugybės vartotojų bei vartotojų grupių struktūra ir administravimas, taip pat priėjimas prie duomenų prisijungus nuotoliniu būdu arba redaguojant scenarijus;

- klientų, turinčių daugialypiai susijusių duomenų šaltinius, sinchronizavimas ir, pagal pareikalavimą, failų bei taikomųjų programų atsisiuntimas.

Įrenginiams, turintiems mažai resursų, tokiems kaip mobilieji telefonai arba pranešimų gavikliai, DB2 Everyplace Sync Server siūlo Java 2 Micro Edition (J2ME) Sync Client. J2ME Sync Client (sinchronizavimo klientas) tinka mobilios informacijos įrenginio profiliui (MIDP) ir siūlo žymiausio standarto platformą įrenginių projektavimui mobilių duomenų rinkoje.

3.5.2 Lotus EasySync Pro

Lengvai instaliuojamas ir naudojamas IBM Lotus EasySync Pro leidžia greitai ir nesudėtingai sinchronizuoti duomenis tarp IBM Lotus Notes asmeniniame kompiuteryje ir bevielių įrenginių, kuriuose įdiegta Palm ar Microsoft Windows CE/Pocket PC operacinės sistemos. Grįžus prie savo asmeninio kompiuterio ir atlikus nesudėtingą sinchronizavimo procesą, galima greitai atnaujinti asmeninę informaciją.

EasySync Pro siūlo naujas efektyvias sinchronizavimo ypatybes, tokias kaip pagerinta duomenų filtravimo technologija ir patobulintos konfigūracijos pasirinktys. Į plačios EasySync Pro konfigūracijos parinktis įeina laukų paskirstymas iš Lotus Notes į mobilųjį įrenginį, o tai padeda užtikrinti duomenų tikslumą.

Pagrindinės funkcijos: [27]

- greita ir nesudėtinga duomenų sinchronizacija tarp Lotus Notes asmeniniame kompiuteryje ir mobiliųjų įrenginių, kuriuose įdiegta Palm ar Microsoft Windows CE/Pocket PC operacinės sistemos;
- priėjimas prie svarbiausios Lotus Notes technologija paremtos informacijos iš bet kur;
- greita programinės įrangos instaliacija ir konfigūracija kompiuteryje;
- sinchronizuojant duomenis suranda Lotus Notes duomenų buvimo vietas ir automatiškai šias vietas pažymi kaip numatytąsias (pagrindines);
- paprastas ir greitas asmeninės informacijos atnaujinimas;
- dvikryptis sinchronizavimas;
- efektyvios sinchronizavimo ypatybės, pvz., pagerinta duomenų filtravimo technologija ir patobulintos konfigūracijos pasirinktys tam, kad būtų pasiektas didesnis lankstumas ir patogumas;
- galimybė nuspręsti, kokie duomenys ir kur bus saugojami – Lotus Notes asmeniniame kompiuteryje ar mobiliajame įrenginyje.

3.6 Išvados

Java ypač tinkama kalba tinkliniam ir serveriniam programavimui: sukurta C ir C++ pagrindu, priešingai nei pastarosios kalbos, Java užtikrina duomenų saugumą, turi standartines bibliotekas tinklinių programų kūrimui. Be to, Java 2 Platform Micro Edition (J2ME) – patogi platforma programų mobiliems įrenginiams projektavimui, todėl Java palaikymas integruotas daugelyje šiuolaikinių mobiliųjų įrenginių.

Java technologijomis pagrįsta IBM kompanijos WebSphere programinių produktų šeima – galingas ir universalus įrankis tinklo aplikacijų kūrimui ir palaikymui, o Everyplace, Transcoding ir Voice produktų serijos suteikia nepamainomų galimybių aplikacijų mobiliems įrenginiams kūrėjams.

Sistemose, skirtose mobiliajai prieigai ir sukurtose DB2 Everyplace duomenų bazių valdymo sistemos pagrindu, duomenų sinchronizavimui patogiu naudoti DB2 Everyplace Sync Server.

4 Programų kūrimo priemonių analizė

4.1 Įvadas

Programų mobiliesiems įrenginiams plėtojimas per pastaruosius keletą metų labai išaugo. Svarbu tai, jog tradicinis požiūris į tokių programų vystymą netinka – būtina taikytis prie ribotų mobilaus įrenginio galimybių: atminties kiekio, procesoriaus galios bei nedidelio įrenginio ekrano. Beje, sparčiai tobulėjant mobiliųjų įrenginių techninėms galimybėms, šios problemos tampa vis paprasčiau išsprendžiamos.

Paprastai, tam, kad sukurtume programą mobiliajam įrenginiui, reikalingas visas kompleksas skirtingos paskirties priemonių: programavimo platforma, duomenų bazių valdymo sistema bei duomenų sinchronizavimo priemonė, patogūs įrankiai programos kodo rašymui. Skirtingi gamintojai (IBM, Nokia, Metrowerks ir kt.) siūlo gana panašių galimybių programinius produktus, todėl vienas kriterijų, lemiančių pasirinkimą, galėtų būti produktų tarpusavio suderinamumas ir integracija. 3-iaame skyriuje kalbėjome apie Java kalbos ir Java technologijomis pagrįstos WebSphere šeimos produktų tinkamumą programų mobiliesiems įrenginiams vystymui. Šiame skyriuje išanalizuosime konkretų Sun Microsystems ir IBM kompanijų siūlomą produktų rinkinį, optimaliai tinkantį tinklinių programų mobiliems įrenginiams kūrimui ir palaikymui.

4.2 Programų kūrimo priemonių analizė

4.2.1 Java 2 Micro Edition (J2ME)

Kompanija Sun Microsystems siūlo Java programavimo kalbos platformą J2ME (Java 2 Micro Edition), sukurtą specialiai mobiliesiems įrenginiams. Iš esmės tai „apkarpytas“ standartinės Java platformos (Java 2 Standard Edition) variantas, sukurtas atsižvelgiant į visus mobiliųjų įrenginių apribojimus (nedidelę atmintį, ribotas procesoriaus galimybes, grafinės sąsajos ir informacijos įvedimo bei išvedimo ypatumus, nepastovų sujungimą su mobiliojo ryšio tinklu ir t.t.). Beleidė Java, naudojama mikroprocesorių kortelėse, pranešimų gavikliuose, mobiliuosiuose telefonuose, komunikatoriuose, automobilinėse telemetrijos sistemose ir delniniuose kompiuteriuose, užtikrina nepaprastą šių įrenginių lankstumą ir padeda vartotojui veiksmingiau dirbti. Be to, lyginant su kitomis programavimo aplinkomis, Java platforma turi labai svarbų privalumą: ji beveik nepriklauso nuo programinės ir techninės įrenginių įrangos. Vienintelė sąlyga – prietaisas turi būti suderinamas su virtualiąja Java mašina (Java Virtual Machine, JVM), kuri be „tarpininkų“ vykdo visas Java kalba parašytas programas.

Tačiau, net ir naudojantis J2ME galimybėmis, viena sistema netinka visiems įrenginiams. Todėl J2ME platforma yra suskaidyta į konfigūracijas ir profilius, kurie suteikia specifinę informaciją apie tam tikrą grupę mobilių įrenginių. CDC (Connected Device Configuration) – konfigūracija, skirta sudėtingesniems mobiliems įrenginiams, tokiems kaip pvz. komunikatoriai. CLDC (Connected Limited Device Configuration) – konfigūracija paprastesniems įrenginiams, tokiems kaip mobilieji telefonai ar delniniai kompiuteriai. MIDP (Mobile Information Device Profile) – profilis, skirtas CLDC paremtiems mobiliems įrenginiams.

3.1 pav. stačiakampiu apibraukta šiame darbe nagrinėjama mobiliųjų technologijų sritis. Ji apima specialų Java kalbos kompiliatorių mažiems įrenginiams (K Virtual Machine, KVM), pačią Java programavimo kalbą, CLDC konfigūraciją bei MIDP profilį.

CLDC konfigūracija [28]. Konfigūracijos tikslas – standartizuoti labai portatyvią, kuo mažiau resursų užimančią programų vystymo platformą, skirtą mobiliams įrenginiams. Tokia aplinka suteikia galimybę dinamiškai atsisiųsti programą bei duomenis į mobilųjį įrenginį, taip pat galimybę kurti ir plėtoti trečios šalies programinę įrangą.

CLDC konfigūracija apima standartines java.* klases, kurios yra tokios pat kaip ir J2SE platformos versijoje:

- 16 klasių iš java.lang paketo;
- 13 – iš java.io;
- 8 – iš java.util;
- 29 klaidų ir išimčių klasės.

Be to, yra ir vienas paketas specialių CLDC klasių – javax.microedition.io. Standartinės bibliotekos, susijusios su tinklais, įvedimu – išvedimu, CLDC konfigūracijai yra per didelės, todėl yra pateikiama nauja grupė klasių (Generic Connection framework), kuri aprašo tik pagrindinius sujungimo tipus. Konkretūs protokolai yra aprašomi profiliuose.

MIDP profilis [29]. MIDP – Java technologija paremtų vartotojo sąsajos kūrimo priemonių rinkinys, papildantis CLDC konfigūraciją ir suteikiantis daugiau galimybių mobiliams įrenginiams, tokiems kaip mobilūs telefonai ar delniniai kompiuteriai. Profilis praplečia programų lankstumą, taikant jas įvairiems įrenginiams, neprarandant pagrindinės mobiliųjų technologijų savybės – portatyvumo.

Profilis apima [16]:

- ekrano valdymą, įvedant duomenis;
- nuolatinį duomenų saugojimą;
- duomenų siuntimą (trumposios žinutės, elektroninis paštas ir t.t.);
- saugumą ir veikimą per bevielių tinklų sujungimus.

Profilyje yra apibrėžtos šios funkcijos [16]:

- programos instaliacija ir gyvavimo ciklo valdymas;
- vartotojo sąsajos palaikymas;
- įvykių apdorojimas;
- aukšto lygio programos modelis;
- duomenų bazių palaikymas.

MIDP apibūdina šias klases iš paketo javax.microedition.*:

- javax.microedition.midlet (programos gyvavimo ciklas);
- javax.microedition.lcdui (vartotojo sąsaja);
- javax.microedition.rms (nuolatinis duomenų buvimas);
- javax.microedition.io (tinklai);
- java.lang ir java.util (kalba ir papildomos funkcijos).

Programa, parašyta remiantis MIDP profiliu, yra vadinama MIDletu. Tai programa, kuri paleidžiama ir sustabdoma pagal pareikalavimą, veikia naudodama vidinius mobiliojo įrenginio resursus, paprastai saugoma pastoviojoje mobiliojo įrenginio atmintyje bei gali būti paleidžiama įrenginyje ne vieną kartą.

4.2.2 DB2 Everyplace

IBM kompanijos produktas – duomenų bazių valdymo sistema, skirta mobiliesiems įrenginiams. Tai visapusė kūrimo terpė, turinti instrumentus programų projektavimui, vystymui ir palaikymui bei aukšto efektyvumo duomenų sinchronizavimo mechanizmą.

DB2 Everyplace sudaro trys pagrindiniai komponentai: duomenų bazės variklis, veikiantis mobiliajame įrenginyje, sinchronizavimo serveris (DB2 Everyplace Sync Server, žr. 3.5 poskyrį) bei pagrindinė duomenų bazė, esanti asmeniniame kompiuteryje arba organizacijos serveryje. Vartotojas gali įsidėti pagrindinės duomenų bazės fragmentus į savo mobilųjį įrenginį ir sinchronizuoti juos su pagrindine duomenų baze. DB2 Everyplace gali būti naudojama ir kaip vietinė nepriklausoma mobilaus įrenginio duomenų bazė. [30]

4.1 lentelėje atlikta DB2 Everyplace galimybių analizė:

4.4 lentelė DB2 Everyplace galimybių analizė

Kriterijus	DB2 Everyplace galimybės
Platforma	Linux, Neutrino, PalmOS, Symbian, Windows, Windows CE, Windows PocketPC
Atminties sunaudojimas	137 Kb
Duomenų bazės dydis	Ribojamas mobilaus įrenginio laisvos atminties kiekiu

Duomenų bazės tipas	Reliacinė
Ryšys su kitomis DB	Per DB2 Everyplace Sync Server su DB2, Oracle, Microsoft SQL server, Domino and Exchange
Sinchronizavimo protokolas	SyncML
Sinchronizavimo konfliktų sprendimas programiškai	Yra
Sinchronizavimo konfliktų sprendimo vieta	Sisteminė ir programuojama
Sinchronizavimas iš kelių serverių	Nėra
Sinchronizavimo stebėjimas	Nėra
Duomenų bazių kiekis	Neribojamas
Lentelių kiekis	Neribojamas
Laukų kiekis	Neribojamas
Palaikomi duomenų tipai	DB2 tipai
Duomenų bazės valdymo įrankiai	Yra
Programavimo priemonės	JDBC, ADO.NET
Saugumas	Autentifikavimas, autorizavimas, šifravimas
Kaina	Nemokama

4.2.3 WebSphere Studio Device Developer

IBM kompanijos produktas – programinė įranga su integruota plėtojimo aplinka (Integrated Development Environment (IDE)), leidžiančia kurti, testuoti bei talpinti J2ME technologija paremtas programas, veikiančias mobiliuose įrenginiuose. Šios programinės įrangos versijos yra sukurtos tiek Windows, tiek Linux operacinėms sistemoms. Kadangi šioje programinėje įrangoje yra įdiegta speciali aplinka WebSphere Micro Environment, tai WebSphere sistema parašytos programos veikia daugelyje šiuo metu rinkoje esančių mobilių įrenginių bei daugelyje operacinių sistemų. WebSphere Studio Device Developer apjungia savyje WebSphere sistemos galimybes ir mobilių įrenginių patogumą.

Integruota plėtojimo aplinka apima šias galimybes: [31]

- SmartLinker™ technologija, kuri optimizuoja programų veikimą, kad šios efektyviai naudotų atmintį, esančią mažuose įrenginiuose;
- įvairių įrenginių ir emuliatorių palaikymas, taip pat ir tų emuliatorių, kurie palaiko Vieningos Emuliatorių Vartotojų Sąsajos (Unified Emulator Interface (UEI)) standartą;
- palaiko IBM kompanijos produktą Service Mangement Framework™ (SMF), kuris valdo programų, perduodamų tinklais, gyvavimo ciklą;
- klaidų suradimas ir ištaisymas bei programos suderinimas, leidžiantis programuotojui testuoti programą, veikiančią kaip realiame mobiliame įrenginyje;
- dinaminė instaliacija integruojant sistemą su tokiais produktais kaip WebSphere Studio Application Developer, WebSphere Studio Site Developer ir Nokia Development Suite.

WebSphere Studio Device Developer savyje turi tekstinį redaktorių, leidžiantį rašyti programos kodą. Nuo kitų tekstinių redaktorių ši programa skiriasi tuo, jog turi daug papildomų funkcijų: grafinį

klasių, bibliotekų ir resursų atvaizdavimą, tam tikrų kodo dalių šablonus (pvz. komentarai, ciklai, duomenų tipai ir t.t.), patogų klasių ir kitų failų bei paketų kūrimą ir kt. Ši programinė įranga taip pat leidžia patogiai rasti ir taisyti klaidas, paleisti programą emuliacijoje, kuris imituoja tikrą mobilųjį įrenginį. Be to, yra galimybė integruoti ir kitokius emuliacorius, kurių galima atsisiųsti iš interneto. [31]

4.2.4 Rational Application Developer

Rational Application Developer (ankstesnė versija vadinosi WebSphere Studio Application Developer) – IBM kompanijos produktas, optimizuotas WebSphere programinei įrangai, tačiau gali būti diegiamas ir kitose vykdymo (runtime) platformose. Padeda kūrėjams greitai projektuoti, kurti, analizuoti, testuoti, profiliuoti ir diegti aukštos kokybės tinklo, į paslaugas orientuotos architektūros (service-oriented Architecture, SOA), Java, J2EE ir portalo aplikacijas. Šis įrankis padeda padidinti produktyvumą, sumažinti patirties kreivę ir sutrumpinti kūrimo ir testavimo ciklus. Jam būdingas pilnas J2EE programavimo modelio palaikymas, integruotos portalo kūrimo funkcijos, Unified Modeling Language™ (unifikuota modeliavimo kalba) vaizdinės redagavimo galimybės, kodo analizės funkcijos ir automatizuoti testavimo ir diegimo įrankiai.

Pagrindinės funkcijos: [32]

- apima vizualaus konstravimo įrankius, skirtus SOA aplikacijoms;
- nesudėtingai kuriamos, testuojamos ir diegiamos J2EE aplikacijos, pilnas J2EE programavimo modelio palaikymas, įskaitant Web, Java, web servisus ir EJB (Enterprise JavaBeans) kūrimą; galimybė vizualizuoti ir grafiškai redaguoti kodą, naudojant UML (Unified Modeling Language) paremtą vizualų redaktorių;
- pilnas rinkinys vizualių portalo kūrimo įrankių, skirtų portletų ir ištisu portalų kūrimui bei derinimui; vedlių pagalba galima kurti visiškai naujus portletus ir portalo projekto struktūras, kurios dera su IBM, J2EE ir JSR (Java Specification Request) specifikacijomis; testavimo serverio dėka testavimui nereikalinga prieiga prie tiesioginio serverio, o tai padeda pagreitinti aukštos kokybės aplikacijų kūrimą;
- automatizuoti diegimo įrankiai apima vietinio ir iš serverio pusės kodo testavimo ir derinimo palaikymą IBM WebSphere Application Server, WebSphere Application Server-Express, WebSphere Portal ir Apache Tomcat sistemose.

Nors tiesioginė Rational Application Developer paskirtis nėra aplikacijų mobiliems įrenginiams kūrimas, tačiau tam tikrose situacijose produktas gali būti sėkmingai panaudotas, pvz. servletų, atsakančių į mobiliojo įrenginio aplikacijos užklausa, kūrimui.

4.3 Išvados

Programų mobiliems įrenginiams optimizavimo uždavinį dalinai išsprendžia programuotojo pasirinkti programavimo įrankiai – J2ME platformai sukurtos programos kompaktiškos, dinamiškai atsisiunčiamos ir lengvai diegiamos.

Pasirenkant programinę įrangą svarbu atsižvelgti į skirtingų produktų tarpusavio suderinamumą ir integraciją. J2ME platformai puikiai tinka IBM WebSphere programavimui skirti įrankiai – WebSphere Device Developer bei Rational Application Developer. Pastarasis produktas nors ir nėra tiesiogiai skirtas programavimui mobiliems įrenginiams, tačiau gali būti panaudotas servletų kūrimui. Tinklinių programų kūrimui, kuomet reikalinga duomenų bazė, šalia jau minėtų produktų dera DB2 Everyplace duomenų bazių valdymo sistema kartu su DB2 Everyplace Sync Server – įrankiu duomenų sinchronizavimui.

Šiame skyriuje išanalizuoti produktai – optimalus sprendimas mobilios el. dienyno realizacijai.

5 Mobilaus el. dienyno modelio realizavimas

5.1 Įvadas

Mobilaus el. dienyno paskirtis nėra pakeisti įprasto el. dienyno formos. Visus mokyklos poreikius tenkinantis el. dienynas – sudėtingas, daugialypis ir nuolat atnaujinamas produktas, todėl daugelio sudėtingesnių veiksmų, tokių kaip didesnių sąrašų apdorojimas, didesnio duomenų kiekio įvedimas ir pan., mobiliuoju įrenginiu atlikti nepatogu. Mobili prieiga prie mokykloje veikiančio el. dienyno galėtų būti tik patogi papildoma paslauga: įvertinimų, tvarkaraščio, pamokų temų peržiūrai, pavienių įvertinimų įrašymui ir pan.

Kadangi šio darbo tikslas išanalizuoti mobilaus el. dienyno kūrimo technologijas, kurios galėtų būti pritaikytos ir panašių tinklinių programų kūrimui, apsiribosime tik produkto modelio realizavimu – sukursime mobiliesiems telefonams orientuotą sistemą, atliekančią mobilaus el. dienyno funkcijas.

Modelio realizavimas atliekamas keliais etapais:

- mobilaus el. dienyno specifikacijų analizė;
- programinės įrangos diegimas (programinės įrangos pasirinkimą pagrindėme 4-ame skyriuje);
- duomenų bazės projektavimas ir kūrimas;
- serverinės dalies (servleto) kūrimas;
- klientinės dalies (MIDleto) kūrimas;
- sistemos testavimas;
- galutinio modelio stovio aprašymas ir tobulinimo perspektyvų numatymas.

5.2 Mobilaus el. dienyno specifikacijos

Darbo pradžioje svarbu aiškiai ir nedviprasmiškai apibrėžti kuriamam programinės įrangos produktui keliamus reikalavimus. Tikslus reikalavimų specifikavimas padeda struktūrizuoti darbo eigą bei užtikrina, jog galutinis produktas atitiks programuotojo lūkesčius. Lygiai taip pat svarbu numatyti sistemos architektūrą, duomenų srautų judėjimą, suprojektuoti vartotojo sąsają.

5.2.1 Reikalavimai sistemai

Funkciniai reikalavimai:

- galimybė naudotis realizuota programa bet kuriame mobiliajame įrenginyje, kuris gali vykdyti sukompilijuotas J2ME platformoje parašytas programas;

- duomenų saugumo užtikrinimas – galimybę prisijungti prie sistemos turi tik registruoti vartotojai, identifikuojami vartotojo vardu ir slaptažodžiu;
- galimybė mokiniui peržiūrėti savo pamokų tvarkaraštį;
- galimybė mokiniui peržiūrėti dalykų įvertinimus;
- galimybė mokiniui peržiūrėti pasirinkto dalyko pamokų temas;
- galimybė mokytojui peržiūrėti savo pamokų tvarkaraštį;
- galimybė mokytojui peržiūrėti mokinių grupių sąrašus;
- galimybė mokytojui įrašyti, peržiūrėti ir redaguoti mokinių įvertinimus;
- galimybė mokytojui įrašyti, peržiūrėti ir redaguoti pamokų temas.

Programiniai reikalavimai:

- mobiliajame įrenginyje turi būti įdiegta J2ME platforma;
- serveryje turi būti įdiegta DB2 duomenų bazių valdymo sistema, ir WebSphere Application Server.

Reikalavimai aparatinei įrangai:

WebSphere Application Server 6.1 (rekomenduojama) palaikymui Microsoft® Windows 2000, Windows 2003, Windows XP2, Linux x86 operacinėse sistemose reikalingi parametrai: [36]

- Intel Pentium ar AMD Opteron 500MHz arba greitesnis procesorius;
- 512MB darbinės atminties (rekomenduojama 1GB);
- 1GB laisvos disko atminties programos instaliavimui;
- CD-ROM įrenginys.

WebSphere Application Server 5.1 (naudojama) palaikymui Windows® 2000, Windows 2003, Linux x86 operacinėse sistemose reikalingi parametrai: [35]

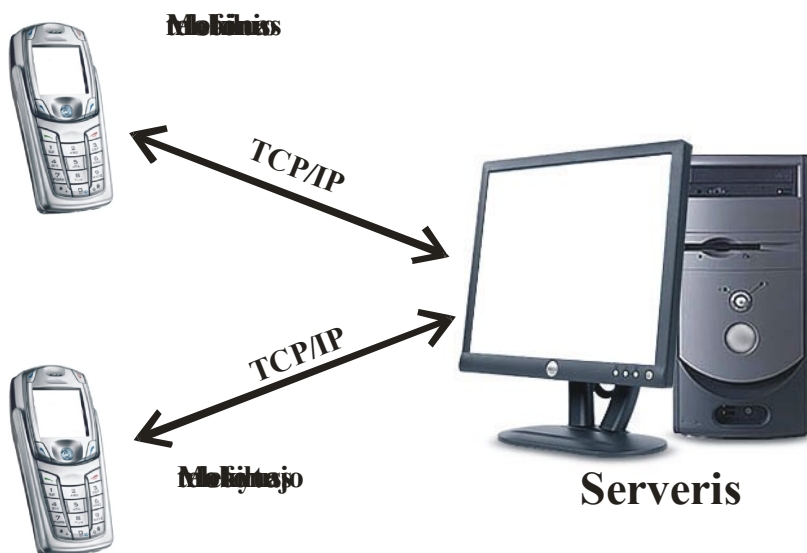
- Intel Pentium, AMD Opteron 500 MHz ar greitesnis procesorius;
- 596 MB laisvos disko atminties programos instaliavimui;
- 256 MB darbinės atminties (rekomenduojama 512 MB);
- CD-ROM įrenginys.

DB2 Universal Database v8.1 palaikymui Windows šeimos operacinėse sistemose reikalingi parametrai: [34]

- Pentium ar panašus procesorius;
- 256 MB darbinės atminties;
- 100-350 MB laisvos disko atminties instaliavimui;

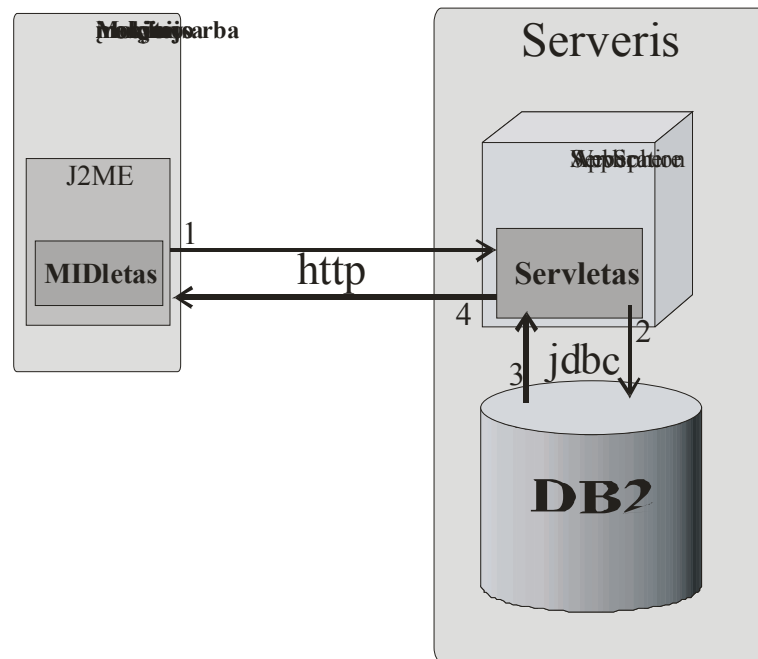
5.2.2 Sistemos architektūra

Mobilaus el. dienyno sistemos veikimo principas pagrįstas kliento-serverio architektūra. Tokią sistemą sudaro serveris, atsakingas už informacijos nuolatinį perdavimą vartotojui ir priėmimą iš vartotojo bei klientas, priimančias serverio jam persiunčiamą informaciją, pateikiantis ją vartotojui, taip pat persiunčiantis vartotojo pateikiamą informaciją serveriui. Sistemos architektūra pavaizduota 5.1 pav.:



5.6 pav. Sistemos architektūra

Kuriamoje programinėje įrangoje aiškiai išskiriamos dvi posistemės: serverio programinė įranga ir kliento programinė įranga. Duomenų srautų judėjimas sistemoje pavaizduotas 5.2 pav.:

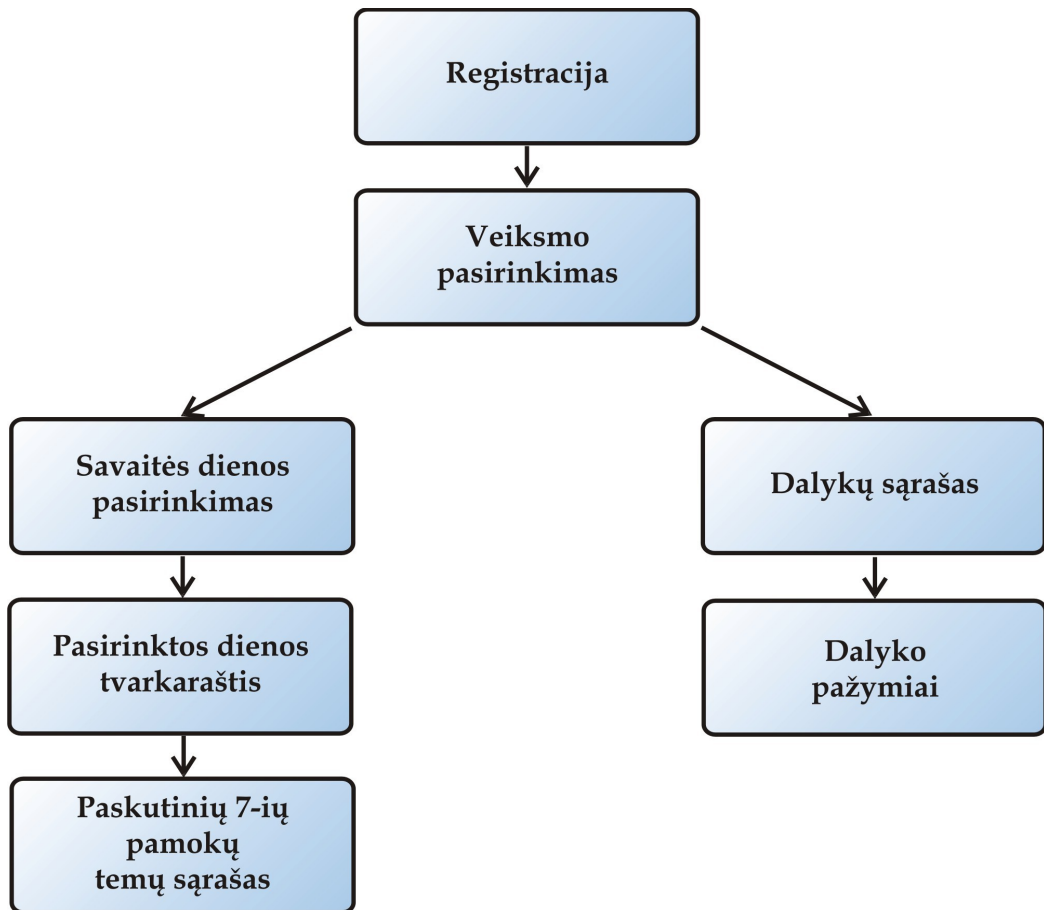


5.7 pav. Duomenų perdavimo schema

- 1 – Vartotojo mobiliajame įrenginyje vykdomas MIDletas HTTP protokolu kreipiasi į serveryje esantį servletą;
- 2 – servletas JDBC (Java Database Connectivity) jungtimi pateikia užklausą DB2 duomenų bazei;
- 3 – įvykdžiusi užklausą, DB2 duomenų bazė užklauskos rezultatus perduoda servletui;
- 4 – servletas užklauskos rezultatus perduoda MIDletui.

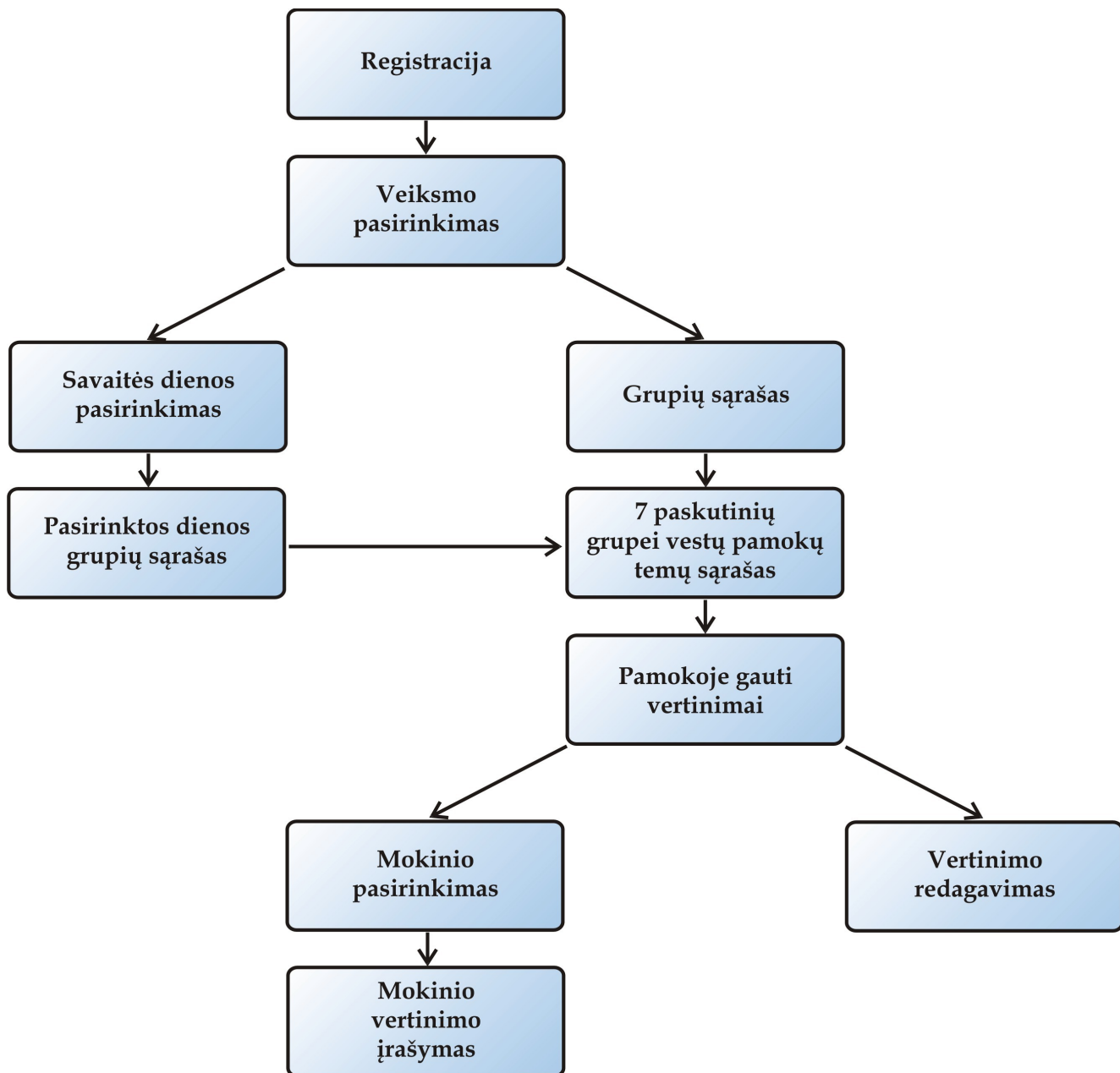
5.2.3 Vartotojo sąsajos projektavimas

Mokinio klientinės dalies vartotojo sąsajos schema pavaizduota 5.3 pav.:



5.8 pav. Mokinio klientinės dalies vartotojo sąsajos schema

Mokytojo klientinės dalies vartotojo sąsajos schema pavaizduota 5.4 pav.:

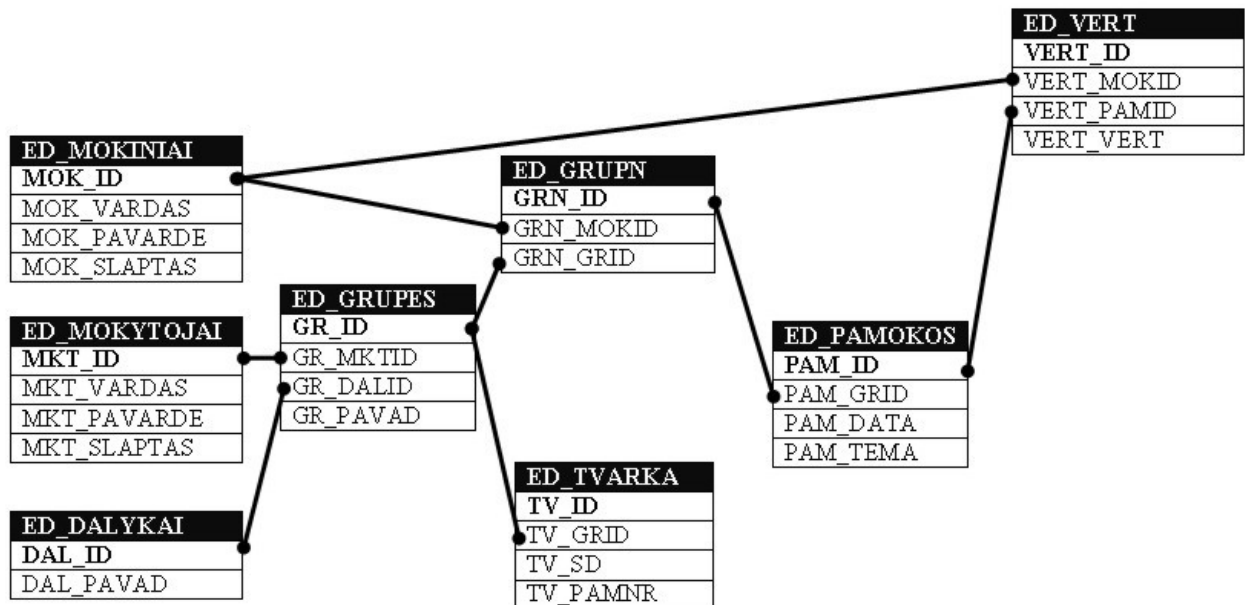


5.9 pav. Mokytojo klientinės dalies vartotojo sąsajos schema

5.3 Mobilaus el. dienyno realizavimas

5.3.1 Duomenų bazės projektavimas ir kūrimas

Duomenų bazė sukurta naudojantis DB2 duomenų valdymo sistemos įrankiu Control Center. Duomenų bazėje ELDIEN sukurtos 8 lentelės, kurių tarpusavio sąryšiai pavaizduoti 5.5 pav. Pastorintu šriftu pavaizduoti pirminiai raktai (primary key), linijomis nurodomi išorinių raktų (foreign key) ryšiai. 5.1 lentelėje nurodyti laukų tipai, bei laukų pavadinimų reikšmės.



5.10 pav. Duomenų bazės lentelės ir jų tarpusavio sąryšiai

- ED_MOKINIAI – mokinių lentelė;
- ED_MOKYTOJAI – mokytojų lentelė;
- ED_DALYKAI – mokomųjų dalykų lentelė;
- ED_GRUPES – mokinių grupių lentelė;
- ED_GRUPN – grupių narių lentelė;
- ED_TVARKA – tvarkaraščio duomenys;
- ED_PAMOKOS – vestų pamokų temų lentelė;

– ED_VERT – mokinių vertinimo lentelė.

5.5 lentelė Duomenų bazės laukų tipai

Lauko pavadinimas	Lauko tipas	Reikšmė
MOK_ID, GRN_MOKID, VERT_MOKID	BIGINT	Mokinio ID; grupės nario mokinio ID; mokinio, kuris buvo įvertintas, ID
MOK_VARDAS	VARCHAR	Mokinio vardas
MOK_PAVARDE	VARCHAR	Mokinio pavardė
MOK_SLAPTAS	VARCHAR	Mokinio slaptažodis
MKT_ID, GR_MKTID	BIGINT	Mokytojo ID; grupės mokytojo ID
MKT_VARDAS	VARCHAR	Mokytojo vardas
MKT_PAVARDE	VARCHAR	Mokytojo pavardė
MKT_SLAPTAS	VARCHAR	Mokytojo slaptažodis
DAL_ID, GR_DALID	BIGINT	Dalyko ID; grupės dalyko ID
DAL_PAVAD	VARCHAR	Dalyko pavadinimas
GR_ID	BIGINT	Grupės ID
GR_PAVAD	VARCHAR	Grupės pavadinimas
GRN_ID, PAM_GRID	BIGINT	Grupės nario ID; grupės, kuriai buvo vykdoma pamoka, ID
TV_ID	BIGINT	Tvarkaraščio įrašo ID
TV_SD	SMALLINT	Savaitės diena
TV_PAMNR	SMALLINT	Pamokos numeris tvarkaraštyje
PAM_ID, VERT_PAMID	BIGINT	Pamokos ID; pamokos, kurioje buvo vertinta, ID
PAM_DATA	DATE	Pamokos data
PAM_TEMA	VARCHAR	Pamokos tema
VERT_ID	BIGINT	Vertinimo ID
VERT_VERT	SMALLINT	Vertinimas

5.3.2 Serverinės dalies (servleto) realizacija

WebSphere Studio Application Developer 5.0 pagalba sukurtas servletas EDServlet, gautas išplečiant HttpServlet klasę. Standartiniai servleto metodai *doGet* ir *doPost* (iššaukiami gavus duomenis atitinkamai GET ir POST metodais) iškviečia metodą *doPerform*. Priklausomai nuo parametrų *v* (reiškiančio veiksmą) ir *t* (reiškiančio vartotojo tipą – mokinys ar mokytojas) iškviečiamas atitinkamas metodas. Visuose metoduose pirmiausia suformuojama SQL užklausa, kuri įvykdoma per JDBC jungtį; gauti rezultatai suformuojami atitinkamu formatu ir grąžinami metodui *doPerform*. Užklausiai vykdyti naudojami du metodai: *executeQuery* (naudojamas duomenų paėmimui iš duomenų bazės) ir *executeUpdate* (naudojamas duomenų įrašymui į duomenų bazę). Gavęs duomenis, metodas *doPerform* išsiunčia juos užklausa atsiuntusiam MIDletui.

Programinis servleto kodas pateikiamas 1 priede.

5.3.3 Klientinės dalies (MIDleto) realizacija

Vienas svarbiausių reikalavimų aplikacijai, skirtai mobiliam įrenginiui – jos kompaktiškumas. Todėl klientinė sistemos dalis skiriama į dvi viena nuo kitos nepriklausomas dalis – mokytojo ir mokinio.

Mokinio klientinės dalies realizacija

Mokinio klientinę dalį sudaro du MIDleto failai: informacinis failas EIDienMok.jad ir sukompiliuotų klasių archyvas EIDienMok.jar, kurio dydis 5,2KB. Archyve yra trys klasės: pagrindinė *MokEIDien* ir pagalbinės: *SavDien* ir *cursorius*. *SavDien* klasė – išplėsta javax.microedition.lcdui paketo klasė List: pasirinkimų sąrašė įrašytos savaitės dienos. Klasė *cursorius* turi du metodus: pagrindinį (main) ir viešąjį (public) *askServer*, kurio paskirtis – gautą užklausą GET metodu nusiųsti serveryje esančiam servletui *EDServlet* ir priimti servleto siunčiamą atsakymą. Pagrindinė klasė *MokEIDien* praplečia javax.microedition.midlet paketo *MIDlet* klasę realizuodama *CommandListener* klasės metodus.

MIDletui startuojant, metodas *MokEIDien* sukuria pastoviai naudojamas komandas (*cmSubj*, *cmReg*, *cmCont*, *cmExit*, *cmView*, *cmMain*, *cmWDays*, *cmBack*) ir iškviečia metodą *regForm*, kuris sukuria registracijos formą *fmMain*. Priklausomai nuo to, kuri iš formoje esančių komandų pasirenkama, metodas *commandAction* iššaukia metodą *mainForm* (jei buvo pasirinkta komanda *cmReg*) arba MIDleto darbo baigimo metodą *destroyApp*. Metodas *mainForm* *cursorius* klasės egzemplioriaus *asker* pagalba užklausia DB2 duomenų bazę vartotojo, kurio ID ir slaptažodis buvo nurodyti registracijos formoje *fmMain*, vardo ir pavardės. Jei duomenų apie vartotojo vardą ir pavardę negaunama (t.y. blogai nurodytas vartotojo id arba slaptažodis), metodo *showAlert* pagalba parodomas pranešimas apie blogai įrašytus duomenis ir grįžtama į registracijos formą. Jei vartotojas atpažįstamas, metodo *giveAction* pagalba parodomas veiksmų pasirinkimo sąrašas *actions*, leidžiantis pasirinkti vieną iš dviejų galimų punktų: „Tvarkaraštis“ ir „Pažymiai“. Pasirinkus punktą „Tvarkaraštis“, iškviečiamas metodas *wdList*, kuris sukuria *SavDienos* klasės egzempliorių *weekd*, leidžiantį pasirinkti, kurios savaitės dienos tvarkaraštį norime matyti. Pasirinkus norimą savaitės dieną, iškviečiamas metodas *giveShedule*, kuris *cursorius* klasės egzemplioriaus *asker* pagalba iš duomenų bazės gauna informaciją apie vartotojo pasirinktos dienos tvarkaraštį. Kadangi duomenys iš servleto gaunami tekstinės eilutės pavidalu, metode *giveShedule* jie yra iš koduojami – išskiriami pamokos numeris, grupės, kuriai ta pamoka vedama, identifikacijos numeris ir dalyko pavadinimas. Grupių ID yra surašomi į masyvą *visokieID*, kuris vėliau naudojamas prireikus peržiūrėti pamokų temas. Komandų *cmWDays*, *cmMain* pagalba galima grįžti, atitinkamai, prie norimos savaitės dienos pasirinkimo ir į pagrindinį meniu arba komandos *cmView* pagalba galime iškviešti metodą *showThemes*, kuris, panašiais į metodo *giveShedule*, veiksmiais pateiks septynias vėliausiai vykusius pasirinkto dalyko pamokų temas.

Pagrindinio meniu pasirinkimų sąrašė *actions* pasirinkus punktą „Pažymiai“, iškviečiamas metodas *subjList*, kuris *cursorius* klasės egzemplioriaus *asker* pagalba iš duomenų bazės gauna vartotojo

dalykų, kurių jis mokosi, sąrašą ir jį pateikia *List* tipo objekto *subj* pavidalu. Pasirinkus norimą dalyką, metodas *showMarks* *TextBox* tipo objekto *marks* pagalba pateikia mokinio gautų įvertinimų sąrašą. Jei įvertinimų nėra, parodomas pranešimas „Pažymių nėra“. Pilnas programos kodas pateikiamas 2 priede, vartotojo vadovas – 3 priede.

Mokytojo klientinės dalies realizacija

Mokytojo klientinę dalį sudaro du MIDlet'o failai: informacinis failas *EIDienMkt.jar* ir sukompiliuotų klasių archyvas *EIDienMkt.jar*, kurio dydis 7,6 KB. Archyve yra trys klasės: pagrindinė *MktEIDien* ir pagalbinės: *SavDien* ir *cursorius*. Klasės *SavDien* ir *cursorius* tokios pat kaip mokinio klientinėje dalyje, todėl pakartotinai jų nebeaprašinėsime. Pagrindinė klasė *MktEIDien* praplečia *javax.microedition.midlet* paketo *MIDlet* klasę realizuodama *CommandListener* klasės metodus.

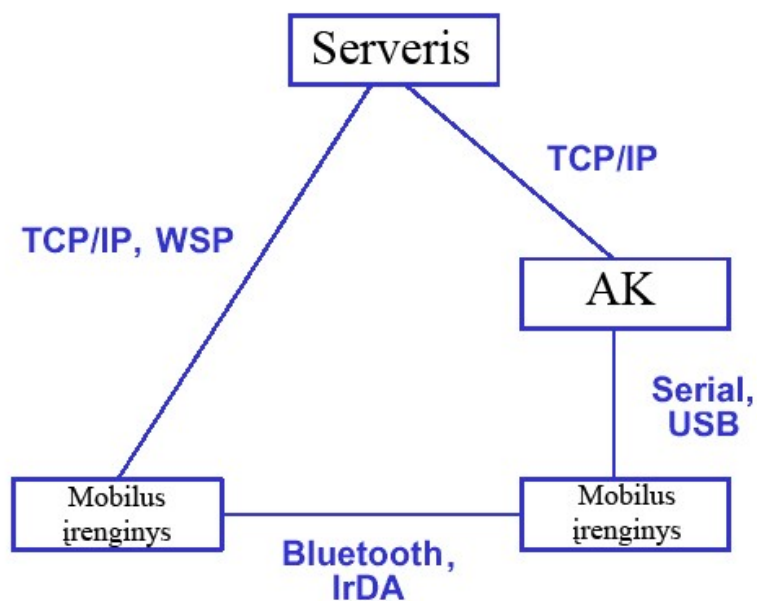
MIDletui startuojant, metodas *MktEIDien* sukuria pastoviai naudojamas komandas (*cmGList*, *cmReg*, *cmCont*, *cmExit*, *cmView*, *cmMain*, *cmWDays*, *cmEditM*, *cmEditT*, *cmSave*, *cmNewM*, *cmThemes*), teksto įvedimo laukelius ir kitus pagalbinius kintamuosius bei iškviečia metodą *regForm*, kuris sukuria registracijos formą *fmMain*. Priklausomai nuo to, kuri iš formoje esančių komandų pasirenkama, metodas *commandAction* iššaukia metodą *mainForm* (jei buvo pasirinkta komanda *cmReg*) arba MIDlet'o darbo baigimo metodą *destroyApp*. Metodas *mainForm* *cursorius* klasės egzemplioriaus *asker* pagalba užklausia DB2 duomenų bazę vartotojo, kurio ID ir slaptažodis buvo nurodyti registracijos formoje *fmMain*, vardo ir pavardės. Jei duomenų apie vartotojo vardą ir pavardę negaunama (t.y. blogai nurodytas vartotojo ID arba slaptažodis), metodo *showAlert* pagalba parodomas pranešimas apie blogai įrašytus duomenis ir grįžtama į registracijos formą. Jei vartotojas atpažįstamas, metodo *giveAction* pagalba parodomas veiksmų pasirinkimo sąrašas *actions*, leidžiantis pasirinkti vieną iš dviejų galimų punktų: „Tvarkaraštis“ ir „Grupės“. Pasirinkus punktą „Tvarkaraštis“, iškviečiamas metodas *wdList*, kuris sukuria *SavDienos* klasės egzempliorių *weekd*, leidžiantį pasirinkti, kurios savaitės dienos tvarkaraštį norime matyti. Pasirinkus norimą savaitės dieną, iškviečiamas metodas *groupListD*, kuris *cursorius* klasės egzemplioriaus *asker* pagalba iš duomenų bazės gauna informaciją vartotojo pasirinktos dienos grupių sąrašą. Grupių ID yra surašomi į masyvą *grupuID*, kuris vėliau naudojamas prireikus peržiūrėti pamokų temas. Komandų *cmWDays*, *cmMain* pagalba galima grįžti, atitinkamai, prie norimos savaitės dienos pasirinkimo ir į pagrindinį meniu arba komandos *cmView* pagalba galime iškviešti metodą *showThemes*, kuris, panašiais į metodo *groupListD*, veiksmiais pateiks septynias vėliausiai vykusius pasirinkto dalyko pamokų temas.

Pagrindinio meniu pasirinkimų sąrašė *actions* pasirinkus punktą „Grupės“, išskviečiamas metodas *groupList*, kuris *cursorius* klasės egzemplioriaus *asker* pagalba iš duomenų bazės gauna vartotojo visų grupių sąrašą ir jį pateikia *List* tipo objekto *groups* pavidalu. Pasirinkus norimą grupę, metodas *showThemes* (kurį galima išskviesti komanda *cmThemes* ir iš grupių sąrašo, gauto pasirinkus meniu punktą „Tvarkaraštis“) *List* tipo objekto *themes* pagalba pateikia paskutinių septynių grupei vestų pamokų sąrašą.

Iš temų sąrašo komandos *cmGList* pagalba galima grįžti prie grupių sąrašo arba, komandos *cmView* pagalba, pagalba peržiūrėti pasirinktos pamokos įvertinimus. Įvertinimus pateikia metodas *showMarks*, objekto *asker* iš duomenų bazės gautus vertinimus pateikdamas *List* tipo objekto *marks* pavidalu. Iš čia galime grįžti prie grupių sąrašo (komanda *cmGList*), temų sąrašo (komanda *cmThemes*), pereiti į pažymio redagavimo sritį (komandos *cmEditM* iššaukiamas metodas *editMark*) ar naujo pažymio įvedimo sritį (komandos *cmNewM* iššaukiamas metodas *newMark*). Redaguoto ir naujo pažymio išsaugojimui atitinkamai naudojami metodai *saveEditedMark* ir *saveNewMark*.

Pilnas programos kodas pateikiamas 2 priede, vartotojo vadovas – 3 priede.

Į mobilųjį telefoną MIDletas gali būti įdiegiamas keliais būdais (žr. 5.6 pav.) [16]: naudojantis TCP/IP (Transmission Control Protocol / Internet Protocol) arba WSP (Wireless Session Protocol) protokolais programą galima parsisiųsti iš serverio. Kitas būdas – atsisiųsti programą į personalinį kompiuterį, naudojantis internetu (TCP/IP protokolu), o tuomet specialiu kabeliu perrašyti programą į mobilųjį įrenginį. Be to, bevielio tinklo duomenų perdavimo metodais (IrDA, Bluetooth) programą galima parsisiųsti iš kito mobiliojo įrenginio.



5.11 pav. MIDleto įdiegimo į mobilių telefoną būdai

5.3.4 Sistemos testavimas

Sistemos testavimui pasirinkome du realius mobiliuosius telefonus (Siemens C55 ir Nokia 3100) ir du mobiliųjų telefonų emuliatorius (Siemens SL 45i ir Nokia 3300).

Sistemos testavimui pasirinkti kriterijai ir testavimo rezultatai:

- **Prieinamumas.** Visuose keturiuose telefonuose programa veikė be sutrikimų.
- **Patogumas diegiant į mobilių įrenginį.** Programą pavyko įdiegti ir parsisiunčiant ją internetu, ir USB kabelio pagalba.
- **Apimties dydis.** Programa naudoja itin mažai mobiliojo telefono atminties resursų: mokinio klientinės dalies supakuotos programos – JAR bylos – dydis 6,8 KB, išpakuotos – 12 KB. Mokytojo klientinės dalies supakuotos programos dydis 7,6 KB, išpakuotos – 14,1 KB.
- **Veikimo greitis.** Veikimo greitis priklauso nuo įrenginio procesoriaus greičio bei ryšio su serveriu greitaveikos.
- **Vartotojo sąsajos patogumas.** Tobulintinas (neatvaizduojami kai kurie lietuvių abėcėlės simboliai su diakritiniais ženklais).

5.3.5 Problemų analizė ir jų sprendimo būdai

Realizuojant mobilaus el. dienyno modelį, iškilo keletas problemų:

1. *Aukšti programinės įrangos reikalavimai kompiuterio aparatinei įrangai.* Naujausių versijų WebSphere Application Server (6.1 versija), WebSphere Studio Device Developer (5.7 versija) ir IBM Rational Application Developer (6.0 versija) sistemos kiekviena reikalauja kompiuterio su 512 MB ir daugiau darbinės atminties. Tačiau dirbti šiomis programomis kompiuteriu, turinčiu tik 512 MB darbinės atminties, buvo beveik neįmanoma, nes darbo metu dažnai reikėdavo pasileisti jas visas kartu. Problemą pavyko išspręsti pradėjus naudotis žemesnės versijos programine įranga, ne tokią reiklią kompiuterio resursams.
2. *Lietuvių abėcėlės simbolių su diakritiniais ženklais atvaizdavimas mobiliajame telefone.* Ne visi simboliai buvo tvarkingai rodomi. Tačiau ši problema susijusi ne su programavimo galimybėmis, o su pačių mobilių įrenginių programine įranga. Nors ir skelbiama, kad Java naudoja Unicode koduotę, kai kurie programos meniu punktu pavadinimuose esantys simboliai buvo vaizduojami neteisingai. Problemų kėlė simboliai „ė“, „į“, „ū“ ir „ų“. Neteisingai vaizduojami ir duomenys, imami iš duomenų bazės. Jei programos meniu punktuose simbolių su diakritiniais galima atsisakyti, tai šiuo atveju toks sprendimas nėra tinkamas, nes duomenys iš duomenų bazės gali būti imami ir vaizdavimui įprastose naršyklėse, kurios problemų su specifinių simbolių vaizdavimu neturi.
3. *Duomenų sinchronizavimas.* Planuojant dirbti su didesniais duomenų kiekiais, būtų tikslinga naudoti duomenų sinchronizavimo priemones. Kadangi darbas su sinchronizavimo priemonėmis reikalauja daug didesnių programavimo bei duomenų bazių administravimo žinių bei įgūdžių, šiame darbe aprašytame el. dienyno modelyje duomenų sinchronizavimo galimybė nerealizuota, tačiau parinkta ir išanalizuota tam skirta programinė įranga. Duomenų sinchronizavimo uždavinį planuojama išspręsti tobulinant sistemą.

5.3.6 Galutinio modelio stovio aprašymas, tobulinimo bei panaudojimo perspektyvų numatymas.

Realizuotos visos dalys, numatytos projektuojant sistemą – serverinė dalis ir mokinio bei mokytojo klientinės dalys. Neišspręstas duomenų sinchronizavimo uždavinys bei lietuviškų rašmenų problema (žr. į 5.3.5 poskyrį). Klientinę dalį galima rasti ir atsisiųsti adresu www.daukantas.lt/emag.

Realizuojant sistemos modelį įgytą patirtį numatoma panaudoti mobiliąja prieiga papildant Šiaulių Simono Daukanto vidurinės mokyklos el. dienyną.

5.4 Išvados

Mobilaus el. dienyno paskirtis – įprasto el. dienyno papildymas mobiliąja prieiga: mobiliuoju telefonu patogiu atlikti tik nesudėtingus veiksmus, t.y. peržiūrėti duomenis, įrašyti ir redaguoti nedidelį duomenų kiekį.

Realizuota klientinė mokinio dalis leido įsitikinti, jog J2ME platformoje sukurtos programos pasižymi maža apimtimi ir didelėmis darbo tinkle galimybėmis. Be to, programa sėkmingai veikia skirtinguose mobiliuose įrenginiuose.

Programos realizacijai pasirinkta programinė įranga pasižymi didelėmis galimybėmis, bet yra reikšti kompiuterio resursams.

Įrenginių programinės įrangos netobulumai neleidžia išspręsti teisingo simbolių su diakritiniais ženklais atvaizdavimo problemos.

6 Išvados

1. Atlikus GSM technologijų, bevielio tinklo technologijų ir mobiliųjų priemonių architektūros analizę, galima teigti:
 - a) bevielio duomenų perdavimo greičiui artėjant prie fiksuoto interneto vartotojams teikiamų greičių ir vis tobulėjant mobiliųjų įrenginių galimybėms, pastarieji vis dažniau pakeis asmenini kompiuterį;
 - b) bevielio tinklo technologijomis perduodamų duomenų greitis nepranoksta įprastų vielinių tinklų, tačiau pastarųjų bus vis dažniau atsisakoma dėl bevielinių tinklų eksploatavimo patogumo.
2. Išanalizavus Java kalbos privalumus ir trūkumus, bei programavimo šia kalba platformų įvairovę, galima daryti išvadą, jog Java ypač tinkama tinkliniam ir serveriniam programavimui, o J2ME platforma patogi programų mobiliems įrenginiams realizavimui.
3. Ištyrus programavimo mobiliems įrenginiams priemonių galimybes bei jų tarpusavio suderinamumą, galima daryti išvadą, jog J2ME platforma, WebSphere Studio Device Developer, Rational Application Developer ir DB2 Everyplace – optimalus įrankių rinkinys mobilaus el. dienyno kūrimui.
4. Sukurtas veikiantis mobilaus el. dienyno modelis (serverinė ir klientinė dalys) bei numatytos jo tobulinimo perspektyvos.
5. Įgyta tinklinių programų projektavimo, programavimo Java kalba ir dokumentavimo patirtis.

7 Literatūra ir informacijos šaltiniai

1. GSM internetinė svetainė: GSM technologijos apžvalga [žiūrėta 2006-03-09]. Prieiga per internetą: <http://www.gsmworld.com/technology/what.shtml>
2. GSM internetinė svetainė: GPRS technologijos apžvalga [žiūrėta 2006-03-09]. Prieiga per internetą: <http://www.gsmworld.com/technology/gprs/index.shtml>
3. GSM internetinė svetainė: EDGE technologijos apžvalga [žiūrėta 2006-03-09]. Prieiga per internetą: <http://www.gsmworld.com/technology/edge/index.shtml>
4. GSM internetinė svetainė: 3GSM technologijos apžvalga [žiūrėta 2006-03-09]. Prieiga per internetą: <http://www.gsmworld.com/technology/3g/index.shtml>
5. Ericsson kompanijos internetinė svetainė: Super 3G technologijos apžvalga [žiūrėta 2006-03-09]. Prieiga per internetą: <http://www.ericsson.com/lt/technology/index.shtml>
6. IrDA Asociacijos internetinė svetainė: IrDA technologijos apžvalga [žiūrėta 2006-01-11]. Prieiga per internetą: <http://www.irda.org/displaycommon.cfm?an=1&subarticlenbr=14>
7. Bluetooth internetinė svetainė: Bluetooth technologijos apžvalga [žiūrėta 2006-01-11]. Prieiga per internetą: <http://www.bluetooth.com/Bluetooth/Learn/Basics/>
8. WiFi internetinė svetainė: WiFi technologijos apžvalga [žiūrėta 2006-01-11]. Prieiga per internetą http://www.wifi.lt/content_2.html
9. WiMax internetinė svetainė: WiMax technologijos apžvalga [žiūrėta 2006-01-11]. Prieiga per internetą <http://www.wimax.com/education>
10. Mobilųjų ryšio priemonių apžvalgos svetainė: mobiliųjų priemonių architektūros apžvalga [žiūrėta 2006-03-13]. Prieiga per internetą <http://www.mobile-review.com/pda.shtml>
11. Mobilaus ryšio operatoriaus Bitė GSM internetinė svetainė: mobiliųjų priemonių architektūros apžvalga [žiūrėta 2006-03-8]. Prieiga per internetą: <http://www.bite.lt/mobilus/>
12. UAB „Infotopas“ internetinė svetainė: delninių kompiuterių architektūros apžvalga [žiūrėta 2006-03-08]. Prieiga per internetą: <http://www.infotop.lt/index.php?page=kainos>
13. Kompanijos IBM internetinė svetainė: mobiliųjų technologijų apžvalga [žiūrėta 2006-03-15]. Prieiga per internetą: <http://www-128.ibm.com/developerworks/wireless/newto/>
14. Enciklopedija Wikipedija: mobiliųjų telefonų istorija [žiūrėta 2006-04-15]. Prieiga per internetą: http://en.wikipedia.org/wiki/History_of_mobile_phones
15. Питер Нортон, Уильям Станек. Программирование на Java. Книга первая. <<СК Пресс>>, 1998 г.
16. Sun Microsystems, Inc. Java Programming for Wireless Devices using J2ME – CLDC / MIDP APIs. Palo Alto, California. 2001

17. Sun kompanijos internetinė svetainė: J2SE platformos apžvalga [žiūrėta 2006-02-10]. Prieiga per internetą: <http://java.sun.com/javase/index.jsp>
18. Sun kompanijos internetinė svetainė: J2EE platformos apžvalga [žiūrėta 2006-02-10]. Prieiga per internetą: <http://java.sun.com/javaee/index.jsp>
19. Sun kompanijos internetinė svetainė: J2ME platformos apžvalga [žiūrėta 2006-02-10]. Prieiga per internetą: <http://java.sun.com/javame/index.jsp>
20. Sun kompanijos internetinė svetainė: Java Card platformos apžvalga [žiūrėta 2006-02-10]. Prieiga per internetą: <http://java.sun.com/products/javacard/>
21. Kompanijos IBM internetinė svetainė: WebSphere Application server šeimos apžvalga [žiūrėta 2006-02-24]. Prieiga per internetą: <http://www-306.ibm.com/software/webservers/appserv/was/>
22. Kompanijos IBM internetinė svetainė: WebSphere Portal apžvalga [žiūrėta 2006-02-24]. Prieiga per internetą: <http://www-306.ibm.com/software/info1/websphere/index.jsp?tab=products/portal>.
23. Kompanijos IBM internetinė svetainė: WebSphere Everyplace šeimos apžvalga [žiūrėta 2006-03-09]. Prieiga per internetą: <http://www-306.ibm.com/software/pervasive/everyplace/>
24. Kompanijos IBM internetinė svetainė: WebSphere Transcoding apžvalga [žiūrėta 2006-03-12]. Prieiga per internetą: http://www-306.ibm.com/software/pervasive/transcoding_publisher/
25. Kompanijos IBM internetinė svetainė: WebSphere Voice šeimos apžvalga [žiūrėta 2006-03-16]. Prieiga per internetą: <http://www-306.ibm.com/software/voice/>
26. Kompanijos IBM internetinė svetainė: DB2 Everyplace Sync Server apžvalga [žiūrėta 2006-04-02]. Prieiga per internetą: <http://www-306.ibm.com/software/data/db2/everyplace/syncserver.html>
27. Kompanijos IBM internetinė svetainė: Lotus EasySync Pro apžvalga [žiūrėta 2006-04-10]. Prieiga per internetą: <http://www-142.ibm.com/software/sw-lotus/easysyncpro>
28. Kompanijos Sun Microsystems internetinė svetainė: konfigūracijos CLDC apžvalga [žiūrėta 2006-04-20]. Prieiga per internetą: <http://java.sun.com/products/cldc/>
29. Kompanijos Sun Microsystems internetinė svetainė: profilio MIDP apžvalga [žiūrėta 2006-04-20]. Prieiga per internetą: <http://java.sun.com/products/midp/>
30. Kompanijos IBM internetinė svetainė: DB2 Everyplace apžvalga [žiūrėta 2006-04-29]. Prieiga per internetą: <http://www-306.ibm.com/software/data/db2/everyplace/>
31. Kompanijos IBM internetinė svetainė: WebSphere Studio Device Developer apžvalga [žiūrėta 2006-05-01]. Prieiga per internetą: <http://www-306.ibm.com/software/wireless/wsdd/>
32. Kompanijos IBM internetinė svetainė: Rational Application Developer apžvalga [žiūrėta 2006-05-05]. Prieiga per internetą: <http://www-306.ibm.com/software/awdtools/developer/application/>
33. R.Belevičius. Java technologijos. Mokomoji knyga. V.: „Technika“, 2005

34. Duomenų bazėms skirtas leidinys www.databasejournal.com: DB2 reikalavimai aparatinei įrangai: [žiūrėta 2006-05-02]. Prieiga per internetą: <http://www.databasejournal.com/features/oracle/article.php/3075071>
35. Kompanijos IBM internetinė svetainė: WebSphere Application Server 5.1 reikalavimai aparatinei įrangai [žiūrėta 2006-05-02]. Prieiga per internetą: <http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg27007312>
36. Kompanijos IBM internetinė svetainė: WebSphere Application Server 6.1 reikalavimai aparatinei įrangai [žiūrėta 2006-05-02]. Prieiga per internetą: <http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg27007491>
37. Darius Šilingas. Java technologijų apžvalga. Konferencijos medžiaga [žiūrėta 2006-04-20]. Prieiga per internetą: http://www.ktu.lt/lt/mokslas/konf05/konf_02/IT2005/Sekc02.pdf

8 Anotacija

Dar ne taip seniai buvusios naujiena Lietuvoje, mobiliosios technologijos vis sparčiau veržiasi į mūsų kasdienį gyvenimą. Buvęs prabanga, mobilusis telefonas šiandien naudojamas ne tik skambinimui bei trumpųjų žinučių siuntimui, bet ir atlieka aibę kitų funkcijų. Bevielių tinklų bei įrenginių technologijos ir tiems įrenginiams pritaikytų paslaugų vystymąsi, ypač pastaraisiais metais, nulėmė tai, jog mobiliųjų technologijų panaudojimas labai išaugo visame pasaulyje. Taigi, mobiliosios technologijos tampa neatsiejama buities, pramogų, verslo, mokslo, politikos dalimi ir Lietuvoje.

Šiame darbe analizuojamos mobilaus el. dienyno kūrimo technologijos, siekiant patobulinti iki šiol naudotą elektroninio dienyno formą – dienynas būtų pasiekiamas bei valdomas ir mobiliųjų įrenginių pagalba. Be to, išanalizuotos technologijos galėtų būti pritaikytos panašių programų, skirtų mobiliems įrenginiams, kūrimui.

Darbe analizuojamos GSM, bevielio tinklo technologijos, mobiliųjų įrenginių architektūra, programų mobiliems įrenginiams kūrimo priemonės. Atliekama Java platformos ir ja pagrįstos WebSphere programinių produktų šeimos bei duomenų bazių sinchronizavimo priemonių analizė. Taip pat aprašomos mobilaus el. dienyno specifikacijos, modelio projektavimo ir realizavimo darbai.

9 Summary

The technologies of mobile e-journal development

Not a long while ago mobile technologies were introduced in Lithuania and now they are quickly spreading in our everyday life. Having been a luxury, a mobile telephone today is being used not only for calling and SMS sending, but it also performs many other functions. The considerable use of mobile technologies all over the world has determined the development of wireless networks and device technologies as well as services fitted for the devices, particularly in the recent years. As a result, mobile technologies are becoming an inseparable part of entertainment, business, education and politics in Lithuania.

This work dwells upon the analysis of mobile e-journal development technologies, seeking to improve the form of e-journal which has been used up to now – the journal could be accessed and managed using mobile devices. Besides, the analyzed technologies could be applied to the development of similar programs for mobile devices.

In the work, GSM, wireless network technologies, architecture for mobile devices and development tools for individual and network applications are analyzed. The analysis of Java platform and Java-based WebSphere software family and tools for the synchronization of databases is also presented. This work also dwells upon the description of e-journal specifications, model development and realization.

10 Priedai

10.1 1 priedas

Servletas EDServlet

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.servlet.Servlet;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class TestServlet extends HttpServlet implements Servlet {

    public TestServlet() {
        super();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        doPerform(request, response);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        doPerform(request, response);
    }

    private void doPerform(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException
    {
        char action = request.getParameter("v").charAt(0);
        String ats = null;
        if (request.getParameter("t") != null)
        {
            String mktid = request.getParameter("mkt_id");
            switch (action)
            {
                case 'r':// Registracija
                    ats = TaccountLookup(mktid, request.getParameter("mkt_slaptas"));
                    break;
                case 'p':// Vienos dienos tvarkaraštis
                    ats = TshowShedule(mktid, request.getParameter("sd"));
                    break;
                case 'g':// Visų grupių sąrašas
                    ats = TgetGroups(mktid);
                    break;
                case 'a':// Vienos dienos grupių sąrašas
                    ats = TgetGroupsD(mktid, request.getParameter("sd"));
                    break;
            }
        }
    }
}
```

```

        case 't':// Temos
            ats = TshowThemes(request.getParameter("gr_id"));
            break;
        case 's':// Redaguoto vertinimo įrašymas
            ats = TsaveEM(request.getParameter("vert_id"),
request.getParameter("p"));
            break;
        case 'n':// Naujo vertinimo įrašymas
            ats = TsaveNewM(request.getParameter("pam_id"),
request.getParameter("p"), request.getParameter("mok_id"));
            break;
        case 'v':// Mokinių įvertinimų sąrašas
            ats = TgetMarks(request.getParameter("pam_id"));
            break;
        case 'd':// Grupės dalyvių sąrašas
            ats = TgetGroupMembers(request.getParameter("pam_id"));
            break;
    }
}
else {
    String mokid = request.getParameter("mok_id");
    switch (action)
    {
        case 'r':// Registracija
            ats = accountLookup(mokid, request.getParameter("mok_slaptas"));
            break;
        case 'p':// Vienos dienos tvarkaraštis
            ats = showShedule(mokid, request.getParameter("sd"));
            break;
        case 'd':// Dalykų sąrašas
            ats = getSubj(mokid);
            break;
        case 't':// Temų sąrašas
            ats = showThemes(request.getParameter("gr_id"));
            break;
        case 'v':// Vertinimų sąrašas
            ats = getMarks(mokid, request.getParameter("dal_id"));
            break;
    }
}
if (ats == null) // jei nieko nerasta
{
    response.sendError(response.SC_BAD_REQUEST, "Blogi duomenys.");
    return;
}
response.setContentType("text/plain");
PrintWriter out = response.getWriter();
out.print(ats);
out.close();
}

//*****
// Mokytojų metodai
//*****
private String TaccountLookup(String id, String slaptas)
{
    Connection con = null;
    Statement st = null;
    String ats = null;
    StringBuffer msgb = new StringBuffer("");
    try
    {
        // randame DB tvarkyklę
        Class.forName ("com.ibm.db2.jcc.DB2Driver").newInstance();

```

```

        // prisijungiame prie DB
        con = DriverManager.getConnection("jdbc:db2://localhost:50000/eldien",
"DbUser", "dbPassw");
        Statement stmt = con.createStatement(); // pasiruošiame jungtį
        // vykdome užklausą
        ResultSet rs = stmt.executeQuery("SELECT mkt_vardas, mkt_pavarde FROM
ed_mokytojai WHERE mkt_id = " + id + " AND mkt_slaptas = '" + slaptas + "'");
        if (rs.next())// jei rastas įrašas
        {
            ats=rs.getString(1)+" "+rs.getString(2); // formuojame atsakymą
            return ats;
        }
        else return null;
    }
    catch (Exception e) // gaudome negerus įvykius :)
    {
        return e.toString();
    }
}

private String TshowShedule(String id, String sd) // dienos grupių sąrašas
{
    Connection con = null;
    Statement st = null;
    String ats = "";
    StringBuffer msgb = new StringBuffer("");
    try
    {
        // randame DB tvarkyklę
        Class.forName ("com.ibm.db2.jcc.DB2Driver").newInstance();
        // prisijungiame prie DB
        con = DriverManager.getConnection("jdbc:db2://localhost:50000/eldien",
"DbUser", "dbPassw");
        Statement stmt = con.createStatement(); // pasiruošiame jungtį
        // vykdome užklausą
        ResultSet rs = stmt.executeQuery("SELECT tv_pamnr, gr_pavad, gr_id FROM
ed_tvarka, ed_grupes "+
"WHERE tv_savd="+sd+" AND tv_grid=gr_id AND gr_mktid=" + id+ " ORDER BY
tv_pamnr");
        while (rs.next()) // kol yra įrašų formuojame atsakymą
        {ats=ats+rs.getString(3)+"*"+rs.getString(1) + ". " +rs.getString(2) +
"|";}
        return ats;
    }
    catch (Exception e)
    {
        return e.toString();
    }
}

private String TgetGroups(String id) // grupių sąrašas
{
    Connection con = null;
    Statement st = null;
    String ats = "";
    StringBuffer msgb = new StringBuffer("");
    try
    {
        // randame DB tvarkyklę
        Class.forName ("com.ibm.db2.jcc.DB2Driver").newInstance();
        // prisijungiame prie DB
        con = DriverManager.getConnection("jdbc:db2://localhost:50000/eldien",
"DbUser", "dbPassw");
        Statement stmt = con.createStatement(); // pasiruošiame jungtį

```

```

// vykdomė užklausa
ResultSet rs = stmt.executeQuery("SELECT gr_pavad, gr_id "+
"FROM ed_grupes WHERE gr_mktid=" + id+ " ORDER BY gr_pavad");
while (rs.next())
    {ats=ats+rs.getString(2)+"*"+rs.getString(1) + "|";}
return ats;
}
catch (Exception e)
{
    return e.toString();
}
}

private String TshowThemes(String gr_id) // temų sąrašas
{
    Connection con = null;
    Statement st = null;
    String ats = "";
    StringBuffer msgb = new StringBuffer("");
    try
    {
        // randame DB tvarkyklę
        Class.forName ("com.ibm.db2.jcc.DB2Driver").newInstance();
        // prisijungiamo prie DB
        con = DriverManager.getConnection("jdbc:db2://localhost:50000/eldien",
"DbUser", "dbPassw");
        Statement stmt = con.createStatement(); // pasirošome jungtį
        // vykdomė užklausa
        ResultSet rs = stmt.executeQuery("SELECT pam_id, pam_data, pam_tema "+
"FROM ed_pamokos WHERE pam_grid="+gr_id+" ORDER BY pam_data DESC FETCH FIRST
7 ROWS ONLY");
        while (rs.next())
            {ats=ats+rs.getString(1)+"*"+rs.getString(2) + " "+ rs.getString(3) + "|";}
        return ats;
    }
    catch (Exception e)
    {
        return e.toString();
    }
}

private String TgetMarks(String pam_id) // pažymiai
{
    Connection con = null;
    Statement st = null;
    String ats = null;
    StringBuffer msgb = new StringBuffer("");
    try
    {
        // randame DB tvarkyklę
        Class.forName ("com.ibm.db2.jcc.DB2Driver").newInstance();
        // prisijungiamo prie DB
        con = DriverManager.getConnection("jdbc:db2://localhost:50000/eldien",
"DbUser", "dbPassw");
        Statement stmt = con.createStatement(); // pasirošime jungtį
        // vykdomė užklausa
        ResultSet rs = stmt.executeQuery("SELECT vert_id, mok_vardas, mok_pavarde,
vert_vert "+
"FROM ed_vert, ed_mokiniai WHERE vert_pamid=" + pam_id
+ " AND vert_mokid=mok_id ORDER BY mok_pavarde, mok_vardas DESC");
        while (rs.next())
            {
                ats=ats+rs.getString(1)+"*"+ rs.getString(2).substring(0,1)+
                "."+rs.getString(3) +

```

```

        " "+rs.getString(4) + " |";
    }
    if (ats=="")
    {return "Pazymiu nera";}
    else { return ats;}
}
catch (Exception e)
{
    return e.toString();
}
}

private String TsaveEM (String vert_id, String vert_vert) // redaguoto pažymio išsaugojimas
{
    Connection con = null;
    Statement st = null;
    String ats = null;
    StringBuffer msgb = new StringBuffer("");
    try
    {
        // randame DB tvarkyklę
        Class.forName ("com.ibm.db2.jcc.DB2Driver").newInstance();
        // prisijungiame prie DB
        con = DriverManager.getConnection("jdbc:db2://localhost:50000/eldien",
"DbUser", "dbPassw");
        String sql = "UPDATE ed_vert SET vert_vert="+vert_vert+" WHERE
vert_id="+vert_id;
        try
        {
            PreparedStatement pstmt = con.prepareStatement(sql);
            pstmt.executeUpdate(); // įvykdome atnaujinimą
            return "gerai";
        }
        catch (SQLException e)
        { return e.toString(); }
    }
    catch (Exception e)
    { return e.toString(); }
}

private String TgetGroupMembers(String pam_id) // grupės mokinių sąrašas
{
    Connection con = null;
    Statement st = null;
    String ats = "";
    StringBuffer msgb = new StringBuffer("");
    try
    {
        // randame DB tvarkyklę
        Class.forName ("com.ibm.db2.jcc.DB2Driver").newInstance();
        // prisijungiame prie DB
        con = DriverManager.getConnection("jdbc:db2://localhost:50000/eldien",
"DbUser", "dbPassw");
        Statement stmt = con.createStatement(); // pasiruošiame jungtį
        // vykdome užklausą
        ResultSet rs = stmt.executeQuery("SELECT mok_id, mok_vardas, mok_pavarde "+
"FROM ed_pamokos, ed_grupn, ed_mokiniai WHERE pam_id=" + pam_id +
" AND grn_mokid=mok_id AND grn_grid=pam_grid ORDER BY mok_pavarde,
mok_vardas DESC");
        while (rs.next())
        {ats=ats+rs.getString(1)+ "*" + rs.getString(2).substring(0,1)
+"."+rs.getString(3) + " |";}
    }
}

```

```

        return ats;
    }
    catch (Exception e)
    { return e.toString(); }
}

private String TsaveNewM (String pamid, String vert, String mokid) // naujo
pažymio išsaugojimas
{
    Connection con = null;
    Statement st = null;
    String ats = null;
    StringBuffer msgb = new StringBuffer("");
    try
    {
        // randame DB tvarkyklę
        Class.forName ("com.ibm.db2.jcc.DB2Driver").newInstance();
        // prisijungiamo prie DB
        con = DriverManager.getConnection("jdbc:db2://localhost:50000/eldien",
"DbUser", "dbPassw");
        String sql = "INSERT INTO ed_vert (vert_mokid, vert_pamid, vert_vert) VALUES
("+ mokid + ", " + pamid + ", " + vert + ")";
        try
        {
            PreparedStatement pstmt = con.prepareStatement(sql);
            pstmt.executeUpdate();
            return "gerai";
        }
        catch (SQLException e)
        {return e.toString(); }
    }
    catch (Exception e)
    { return e.toString(); }
}

//*****
// MOKINIO FUNKCIJOS
//*****
private String accountLookup(String mok_id, String mok_slaptas) //registracija
{
    Connection con = null;
    Statement st = null;
    String ats = "";
    StringBuffer msgb = new StringBuffer("");
    try
    {
        // randame DB tvarkyklę
        Class.forName ("com.ibm.db2.jcc.DB2Driver").newInstance();
        // prisijungiamo prie DB
        con = DriverManager.getConnection("jdbc:db2://localhost:50000/eldien",
"DbUser", "dbPassw");
        Statement stmt = con.createStatement(); // pasiruošiame jungtį
        // vykdomė užklausa
        ResultSet rs = stmt.executeQuery("SELECT mok_vardas, mok_pavarde FROM
ed_mokiniai WHERE mok_id = "
+ mok_id + " AND mok_slaptas = '" + mok_slaptas + "'");
        if (rs.next())
        {
            ats=rs.getString(1)+" "+rs.getString(2);
            return ats;
        }
        else return null;
    }
}

```



```

    catch (Exception e)
    { return e.toString(); }
}

private String getMarks(String mok_id, String dal_id) //pažymiai
{
    Connection con = null;
    Statement st = null;
    String ats = "";
    StringBuffer msgb = new StringBuffer("");
    try
    {
        // randame DB tvarkyklę
        Class.forName ("com.ibm.db2.jcc.DB2Driver").newInstance();
        // prisijungiamo prie DB
        con = DriverManager.getConnection("jdbc:db2://localhost:50000/eldien",
"DbUser", "dbPassw");
        Statement stmt = con.createStatement(); // pasiruošiame jungtį
        // vykdomė užklausa
        ResultSet rs = stmt.executeQuery("SELECT vert_vert, pam_data FROM ed_vert,
ed_grupes, ed_pamokos, ed_dalykai WHERE dal_id="+dal_id+" AND vert_pamid=pam_id AND
pam_grid=gr_id AND dal_id=gr_dalid AND vert_mokid="+mok_id+" ORDER BY pam_data
DESC");
        while (rs.next())
        {ats=ats+rs.getString(2)+" => "+rs.getString(1) + "\n";}
        if (ats=="")
        {return "Pazymiu nera";}
        else {return ats;}
    }
    catch (Exception e)
    {return e.toString();}
}

private String getSubj(String mok_id) // dalykai
{
    Connection con = null;
    Statement st = null;
    String ats = "";
    StringBuffer msgb = new StringBuffer("");
    try
    {
        // randame DB tvarkyklę
        Class.forName ("com.ibm.db2.jcc.DB2Driver").newInstance();
        // prisijungiamo prie DB
        con = DriverManager.getConnection("jdbc:db2://localhost:50000/eldien",
"DbUser", "dbPassw");
        Statement stmt = con.createStatement(); // pasiruošiame jungtį
        // vykdomė užklausa
        ResultSet rs = stmt.executeQuery("SELECT dal_pavad, dal_id "+
"FROM ed_grupes, ed_grupn, ed_dalykai WHERE grn_grid=gr_id AND
dal_id=gr_dalid AND grn_mokid=" + mok_id+ " ORDER BY dal_pavad");
        while (rs.next())
        {ats=ats+rs.getString(2)+"*"+rs.getString(1) + "|";}
        return ats;
    }
    catch (Exception e)
    {return e.toString();}
}

private String showShedule(String mok_id, String sd) // tvarkaraštis
{
    Connection con = null;
    Statement st = null;
    String ats = "";

```

```

StringBuffer msgb = new StringBuffer("");
try
{
    // randame DB tvarkyklę
    Class.forName ("com.ibm.db2.jcc.DB2Driver").newInstance();
    // prisijungiamo prie DB
    con = DriverManager.getConnection("jdbc:db2://localhost:50000/eldien",
"DbUser", "dbPassw");
    Statement stmt = con.createStatement(); // pasiruošiamo jungtį
    // vykdome užklausą
    ResultSet rs = stmt.executeQuery("SELECT tv_pamnr, dal_pavadin, gr_id "+
"FROM ed_tvarka, ed_grupes, ed_grupn, ed_dalykai "+
"WHERE tv_savd="+sd+" AND tv_grid=gr_id AND grn_grid=gr_id AND "+
"dal_id=gr_dalid AND grn_mokid=" + mok_id+ " ORDER BY tv_pamnr");
    while (rs.next())
        {ats=ats+rs.getString(1)+"*"+rs.getString(3) + "/" +rs.getString(2) + "|";}
    return ats;
}
catch (Exception e)
    {return e.toString();}
}

private String showThemes(String gr_id) // temos
{
    Connection con = null;
    Statement st = null;
    String ats = "";
    StringBuffer msgb = new StringBuffer("");
    try
    {
        // randame DB tvarkyklę
        Class.forName ("com.ibm.db2.jcc.DB2Driver").newInstance();
        // prisijungiamo prie DB
        con = DriverManager.getConnection("jdbc:db2://localhost:50000/eldien",
"DbUser", "dbPassw");
        Statement stmt = con.createStatement(); // pasiruošiamo jungtį
        // vykdome užklausą
        ResultSet rs = stmt.executeQuery("SELECT pam_data, pam_tema "+
"FROM ed_pamokos WHERE pam_grid="+gr_id+" ORDER BY pam_data DESC FETCH FIRST
7 ROWS ONLY");
        while (rs.next())
            {ats=ats+rs.getString(1)+"*"+rs.getString(2) + "|";}
        return ats;
    }
    catch (Exception e)
        {return e.toString(); }
}
}

```

10.2 2 priedas

Klasė MokElDien

```
import javax.microedition.lcdui.*;
import javax.microedition.midlet.MIDlet;

public class MokElDien extends MIDlet implements CommandListener
{
    private Display display; // Reference to Display object
    private Form fmMain; // Pagrindinė forma
    //Komandos
    private Command cmSubj; // Dalykų sąrašas
    private Command cmReg; // Registruotis
    private Command cmCont; // Tęsti
    private Command cmExit; // Baigti darba
    private Command cmView; // Peržiūrėti
    private Command cmMain; // Pereiti į pagrindinį meniu
    private Command cmWDays; // Peržiūrėti savaitės dienas
    //Pasirinkimai, sąrašai
    private List actions = null; // Pagrindinio meniu veiksmams
    private SavDienos weekd = null; // Savaitės dienos
    private List subj = null; // Dalykai
    private List shed = null; // Tvarkaraštis
    private List themes = null; // Pamokų temos
    // Teksto laukeliai
    private TextBox marks = null; //Pažymiai
    private TextField tfuID; // Vartotojo ID
    private TextField tfPwd; // Slaptažodis
    public String[] dienos; // Savaitės dienų pavadinimai
    private Alert alError; // Pranešimas apie klaidą
    //Pagalbiniai kintamieji
    private int kurEsam = 0;
    public String ats;
    private String errorMsg = null;
    public String userName = null;
    public String[] visokieID;
    private kursorius asker;
    public MokElDien()
    {
        display = Display.getDisplay(this);
        // Kuriame komandas
        cmReg = new Command("Registruotis", Command.SCREEN, 3);
        cmView = new Command("Peržiūrėti", Command.SCREEN, 2);
        cmMain = new Command("Pagrindinis meniu", Command.ITEM, 4);
        cmExit = new Command("Baigti", Command.EXIT, 5);
        cmWDays = new Command("Dienos", Command.ITEM, 2);
        cmSubj = new Command("Dalykų sąrašas", Command.ITEM, 1);
        // ir visa kita
        tfuID = new TextField("Vartotojo ID:", "", 5, TextField.NUMERIC);
        tfPwd = new TextField("Slaptažodis:", "", 10, TextField.ANY |
TextField.PASSWORD);
        dienos = new String[6];
        dienos[1]="Pirmadienio";
        dienos[2]="Antradienio";
        dienos[3]="Trečiadienio";
        dienos[4]="Ketvirtadienio";
    }
}
```

```

    dienos[5]="Penktadienio";

    // Kuriame registracijos forma
    regForm();
}

public void startApp()
{
    display.setCurrent(fmMain);
}
public void pauseApp()
{ }
public void destroyApp(boolean unconditional)
{ }

public void regForm(){ // Registracijos forma
    fmMain = new Form("Registracija");
    fmMain.addCommand(cmExit);
    fmMain.addCommand(cmReg);
    fmMain.append(tfuID);
    fmMain.append(tfPwd);
    fmMain.setCommandListener(this);
}

public void giveAction(){ // Pagrindinių veiksmų pasirinkimas
    actions= new List("Pasirinkite veiksmą gerb. "+userName, Choice.IMPLICIT);
    actions.addCommand(cmExit);
    actions.addCommand(cmView);
    actions.setCommandListener(this);
    actions.append("Tvarkaraštis", null);
    actions.append("Pažymiai", null);
    display.setCurrent(actions);
    kurEsam = 0;
}

public void wdList(){ // Savaitės dienų pasirinkimas tvarkaraščio rodymui
    weekd = new SavDienos("Pasirinkite savaitės dieną", Choice.IMPLICIT);
    weekd.addCommand(cmExit);
    weekd.addCommand(cmMain);
    weekd.addCommand(cmView);
    weekd.setCommandListener(this);
    display.setCurrent(weekd);
    kurEsam = 1;
}

public void giveShedule(int wd){ // Rodome tvarkaraštį
    display = Display.getDisplay(this);
    try
    {
        ats = "";
        visokieID = new String[20];
        asker = new kursorius();
        ats = asker.askServer("?v=p&mok_id=" + tfuID.getString() + "& " +
            "sd=" + Integer.toString(wd));
        shed = new List (dienos[wd]+" tvarkaraštis", Choice.IMPLICIT);
        String temp="";
        int kiek=0;
        for (int i = 0; i < ats.length(); i++)
        {
            if (ats.charAt(i)=='|')
            {
                shed.append(temp.substring(0,temp.indexOf("*"))+". "
                    + temp.substring(temp.indexOf("/")+1) , null);
                visokieID[kiek] = temp.substring(temp.indexOf("*")+1,temp.indexOf("/"));
            }
        }
    }
}

```

```

        temp = "";
        kiek++;
    }
    else {temp=temp+ats.charAt(i);}
}
shed.addCommand(cmWDays);
shed.addCommand(cmMain);
shed.addCommand(cmExit);
shed.addCommand(cmView);
shed.setCommandListener(this);
display.setCurrent(shed);
kurEsam=2;
}
catch (Exception e)
    { System.err.println("Msg: " + e.toString());}
}

public void subjList(){ // Rodome dalykų sąrašą
display = Display.getDisplay(this);
try
{
    ats = "";
    visokieID = new String[20];
    asker = new kursorius();
    ats = asker.askServer("?v=d&mok_id=" + tfuID.getString());
    subj = new List("Dalykų sąrašas", Choice.IMPLICIT);
    String temp="";
    int kiek=0;
    for (int i = 0; i < ats.length(); i++) // Atskiriame dalykus
    {
        if (ats.charAt(i)=='|')
        {
            visokieID[kiek] = temp.substring(0,temp.indexOf("*"));
            subj.append(temp.substring(temp.indexOf("*")+1), null);
            kiek++;
            temp = "";
        }
        else {temp=temp+ats.charAt(i);}
    }
    subj.addCommand(cmView);
    subj.addCommand(cmMain);
    subj.addCommand(cmExit);
    subj.setCommandListener(this);
    display.setCurrent(subj);
    kurEsam=3;
}
catch (Exception e)
    { System.err.println("Msg: " + e.toString());}
}

public void showMarks(){ // Rodome pažymius
display = Display.getDisplay(this);
try
{
    ats="";
    asker = new kursorius();
    ats = asker.askServer("?v=v&mok_id=" + tfuID.getString()+"&dal_id="+
visokieID[subj.getSelectedIndex()]);
    marks = new TextBox("Pažymių sąrašas", "", 256, TextField.UNEDITABLE);
    marks.insert(ats, 0);
    marks.addCommand(cmSubj);
    marks.addCommand(cmMain);
    marks.addCommand(cmExit);
    marks.setCommandListener(this);
}
}

```

```

        display.setCurrent(marks);
    }
    catch (Exception e)
        { System.err.println("Msg: " + e.toString());}
    }

public void showThemes(){ // Rodome temas
display = Display.getDisplay(this);
try
{
    ats = "";
    asker = new kursorius();
    ats = asker.askServer("?v=t&gr_id=" + visokieID[shed.getSelectedIndex()]);
    themes = new List (shed.getString(shed.getSelectedIndex()), Choice.IMPLICIT);
    String temp="";
    for (int i = 0; i < ats.length(); i++)
    {
        if (ats.charAt(i)=='|')
        {
            themes.append(temp.substring(0,temp.indexOf("*"))+" - "
            + temp.substring(temp.indexOf("*")+1) , null);
            temp = "";
        }
        else {temp=temp+ats.charAt(i);}
    }
    themes.addCommand(cmWDays);
    themes.addCommand(cmMain);
    themes.addCommand(cmExit);
    themes.setCommandListener(this);
    display.setCurrent(themes);
    kurEsam=2;
}
catch (Exception e)
    { System.err.println("Msg: " + e.toString());}
}

public void mainForm(){ //Veiksmų pasirinkimo forma
display = Display.getDisplay(this);
try
{
    asker = new kursorius();
    if (userName==null)
    {
        userName = asker.askServer("?v=r&mok_id=" + tfuID.getString() + "&" +
        "mok_slaptas=" + tfPwd.getString());
    }
    if (userName.substring(0,6).equals("Klaida"))
    { userName=null;
        showAlert(userName);}
    else {giveAction();}
}
catch (Exception e)
    { System.err.println("Msg: " + e.toString());}
}

//Meniu pasirinkimo apdorojimas
public void commandAction(Command c, Displayable s)
{
    if (c == cmReg) // Jei registravosi
    { mainForm();}
    else if (c == cmExit) // Jei baigė darbą
    {
        destroyApp(false);
        notifyDestroyed();
    }
}

```

```

}
else if (c == cmMain) // Jei nori grįžti į pagrindinį meniu
{giveAction();}
else if (c == cmView ) //Jei pasirinkta peržiūra
{
    switch (kurEsam) // Tikriname, kuriame punkte buvo
    {
        case 0:
            if (actions.getSelectedIndex()==0)
            {wdList();}
            else {subjList();}
            break;
        case 1:
            giveShedule(weekd.getSelectedIndex()+1);
            break;
        case 2:
            showThemes();
            break;
        case 3:
            showMarks();
            break;
    }
}
else if (c==cmWDays) // Reikia savaitės dienų sąrašo
{ wdList(); }
else if (c==cmSubj) // Reikia dalykų sąrašo
{ subjList(); }
}

private void showAlert(String msg) // Pranešimas apie negerumą
{
    alError = new Alert("Klaida", msg, null, AlertType.ERROR);
    alError.setTimeout(Alert.FOREVER);
    // Po klaidos parodymo, grįžtame į registracijos formą
    display.setCurrent(alError, fmMain);
}
}

```

Klasė MktEIDien

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class MktEIDien extends MIDlet implements CommandListener
{
    private Display display; // Nuoroda į Display objekta
    private Form fmMain; // Pagrindinė forma
    private Form fmNewTheme; // Naujos temos įrašymo forma
    private Form f = null;
    //Komandos
    private Command cmThemes; // Dalykų sąrašas
    private Command cmReg; // Registruotis
    private Command cmCont; // Tęsti
    private Command cmExit; // Baigti darba
    private Command cmView; // Peržiūrėti
    private Command cmGList; // Peržiūrėti grupių sąrašą
    private Command cmMain; // Pereiti į pagrindinį meniu
    private Command cmWDays; // Peržiūrėti savaitės dienas
    private Command cmEditM; // Redaguoti pažymį
    private Command cmNewM; // Įrašyti naują pažymį
    private Command cmEditT; // Redaguoti pažymį
    private Command cmSave; // Išsaugoti
    //Pasirinkimų sąrašai
    private List actions = null; // Pagrindinio meniu veiksmi
    private SavDienos weekd = null; // Savaitės dienos
    //private List subj = null; // Dalykai
    private List shed = null; // Tvarkaraštis
    private List groupsD = null; // Vienos dienos grupės
    private List groups = null; // Vienos dienos grupės
    private List themes = null; // Pamokų temų sąrašas
    private List marks = null; //Pažymiai
    private List students = null; //Mokiniai
    // Teksto laukeliai
    private TextField tfuID; // Vartotojo ID
    private TextField tfPwd; // Slaptažodis
    private TextField tfMark; // Pažimys
    public String[] dienos; // Savaitės dienų pavadinimai
    private Alert alError; // Pranešimas apie klaidą
    // Pagalbiniai kintamieji
    private int kurEsam = 0;
    private int arNaujas = 0;
    private Kursorius asker;
    public String ats = "";
    private String errorMsg = null;
    public String userName = null;
    public String[] dalykuID;
    public String[] grupiuID;
    public String[] pamokuID;
    public String[] mokiniuID;
    public String[] vertID;
    public DateField lDate;

    public MktEIDien()
    {
        display = Display.getDisplay(this);
        // Kuriame komandas
        cmReg = new Command("Registruotis", Command.SCREEN, 3);
        cmView = new Command("Perziureti", Command.SCREEN, 2);
        cmGList = new Command("Grupes", Command.ITEM, 2);
        cmEditM = new Command("Keisti", Command.SCREEN, 2);
    }
}
```



```

cmEditT = new Command("Redaguoti", Command.SCREEN, 2);
cmCont = new Command("Testi", Command.SCREEN, 1);
cmSave = new Command("Saugoti", Command.SCREEN, 1);
cmMain = new Command("Pagrindinis meniu", Command.ITEM, 4);
cmExit = new Command("Baigti", Command.EXIT, 5);
cmWDays = new Command("Dienos", Command.ITEM, 2);
cmNewM = new Command("Irašyti nauja", Command.ITEM, 1);
cmThemes = new Command("Temos", Command.ITEM, 4);
// ir visa kita
tfuID = new TextField("Vartotojo ID:", "", 5, TextField.NUMERIC);
tfPwd = new TextField("Slaptazodis:", "", 10, TextField.ANY |
TextField.PASSWORD);
lDate = new DateField("Data: ", DateField.DATE);
asker = new Kursorius();
dienos = new String[6];
dienos[1]="Pirmadienio";
dienos[2]="Antradienio";
dienos[3]="Trečiadienio";
dienos[4]="Ketvirtadienio";
dienos[5]="Penktadienio";
dalykuID = new String[20];
grupiuID = new String[25];
pamokuID = new String[25];
mokiniuID = new String[30];
vertID = new String[25];
// Kuriame registracijos forma
regForm();
}

public void startApp()
{
display.setCurrent(fmMain);
}
public void pauseApp()
{ }
public void destroyApp(boolean unconditional)
{ }

public void regForm(){ // Registracijos forma
fmMain = new Form("Registracija");
fmMain.append(tfuID);
fmMain.append(tfPwd);
fmMain.addCommand(cmExit);
fmMain.addCommand(cmReg);
fmMain.setCommandListener(this);
}

public void mainForm(){ //Veiksmų pasirinkimo forma
display = Display.getDisplay(this);
try
{
asker = new Kursorius();
userName = asker.askServer("?v=r&t=1&" + "mkt_id=" + tfuID.getString() + "&" +
"mkt_slaptas=" + tfPwd.getString());
if (userName == null)
{ regForm();}
else {giveAction();}
}
catch (Exception e)
{ System.err.println("Msg: " + e.toString());}
}

public void giveAction(){ // Pagrindinių veiksmų pasirinkimas
actions = new List(userName, Choice.IMPLICIT);

```

```

        actions.append("Tvarkarastis", null);
        actions.append("Grupės", null);
        actions.addCommand(cmExit);
        actions.addCommand(cmView);
        actions.setCommandListener(this);
display.setCurrent(actions);
kurEsam = 0;
}

public void wdList(){ // Savaitės dienų pasirinkimas tvarkaraščio rodymui
weekd = new SavDienos("Pasirinkite diena", Choice.IMPLICIT);
    weekd.addCommand(cmMain);
    weekd.addCommand(cmView);
weekd.setCommandListener(this);
display.setCurrent(weekd);
kurEsam = 1;
}

public void groupListD(){ // Rodome dienos grupių sąrašą
display = Display.getDisplay(this);
try
{
    int sd=weekd.getSelectedIndex()+1;
    ats = asker.askServer("?v=p&t=1&mkt_id=" + tfuID.getString() + "&sd=" + sd);
    groups = new List(dienos[sd]+" grupės:", Choice.IMPLICIT);
    String temp="";
    int kiek=0;
    for (int i = 0; i < ats.length(); i++) // Atskiriame dalykus
        {
            if (ats.charAt(i)=='|')
            {
                groups.append(temp.substring(temp.indexOf('*')+1), null);
                grupiuID[kiek] = temp.substring(0,temp.indexOf('*'));
                kiek++;
                temp = "";
            }
            else {temp=temp+ats.charAt(i);}
        }
    groups.addCommand(cmView);
    groups.addCommand(cmWDays);
    groups.addCommand(cmMain);
    groups.setCommandListener(this);
    display.setCurrent(groups);
    kurEsam=5;
}
catch (Exception e)
{ System.err.println("Msg: " + e.toString());}
}

public void groupList(){ // Rodome grupių sąrašą
display = Display.getDisplay(this);
try
{
    asker = new Kursorius();
    ats = asker.askServer("?v=g&t=1&mkt_id=" + tfuID.getString());
    groups = new List("Grupių sąrašas", Choice.IMPLICIT);
    String temp="";
    int kiek=0;
    for (int i = 0; i < ats.length(); i++) // Atskiriame dalykus
        {
            if (ats.charAt(i)=='|')
            {
                groups.append(temp.substring(temp.indexOf('*')+1), null);
                grupiuID[kiek] = temp.substring(0,temp.indexOf('*'));
            }
        }
}
}

```

```

        kiek++;
        temp = "";
    }
    else {temp=temp+ats.charAt(i);}
}
groups.addCommand(cmView);
groups.addCommand(cmMain);
groups.setCommandListener(this);
display.setCurrent(groups);
kurEsam=5;
}
catch (Exception e)
{ System.err.println("Msg: " + e.toString());}
}

public void showThemes(){ // Rodome temas
display = Display.getDisplay(this);
try
{
    asker = new Kursorius();
    ats = asker.askServer("?v=t&t=1&gr_id=" + grupiuID[groups.getSelectedIndex()]);
    themes = new List (groups.getString(groups.getSelectedIndex()),
Choice.IMPLICIT);
    String temp="";
    int kiek=0;
    for (int i = 0; i < ats.length(); i++)
    {
        if (ats.charAt(i)=='|')
        {
            themes.append(temp.substring(temp.indexOf("*")+1) , null);
            pamokuID[kiek] = temp.substring(0,temp.indexOf('*'));
            temp = "";
            kiek++;
        }
        else {temp=temp+ats.charAt(i);}
    }
    themes.addCommand(cmGList);
    themes.addCommand(cmView);
    themes.setCommandListener(this);
    display.setCurrent(themes);
    kurEsam=4;
}
catch (Exception e)
{ System.err.println("Msg: " + e.toString());}
}

public void showMarks(){ // Rodome pažymius
display = Display.getDisplay(this);
try
{
    asker = new Kursorius();
    ats = asker.askServer("?v=v&t=1&pam_id="+ pamokuID[themes.getSelectedIndex()]);
    marks = new List("Pazymiu sarasas",Choice.IMPLICIT);
    String temp="";
    int kiek=0;
    for (int i = 0; i < ats.length(); i++)
    {
        if (ats.charAt(i)=='|')
        {
            marks.append(temp.substring(temp.indexOf("*")+1) , null);
            vertID[kiek] = temp.substring(0,temp.indexOf('*'));
            temp = "";
            kiek++;
        }
    }
}
}

```

```

        else {temp=temp+ats.charAt(i);}
    }
    marks.addCommand(cmGList);
    marks.addCommand(cmThemes);
    marks.addCommand(cmEditM);
    marks.addCommand(cmNewM);
    marks.setCommandListener(this);
    display.setCurrent(marks);
}
catch (Exception e)
    { System.err.println("Msg1: " + e.toString());}
}

public void editMark(){ // Redaguoti pažymį
    display = Display.getDisplay(this);
    String temp=marks.getString(marks.getSelectedIndex());
    tfMark = new TextField(temp.substring(0,temp.indexOf(" ")),
temp.substring(temp.indexOf(" ")+1), 20, TextField.NUMERIC);
    Form f = new Form("Keisti įvertinimą");
    f.append(tfMark);
    f.addCommand(cmSave);
    f.addCommand(cmThemes);
    f.addCommand(cmGList);
    f.setCommandListener(this);
    arNaujas = 0;
    display.setCurrent(f);
}

public void saveEditedMark(){
    try
    {
        asker = new Kursorius();
        ats = asker.askServer("?v=s&t=1&vert_id="+ vertID[marks.getSelectedIndex()]+
"&p="+tfMark.getString());
        showMarks();
    }
    catch (Exception e)
        { System.err.println("Msg: " + e.toString());}
}

public void selectStudent(){ //Pasirinkti mokinį
    display = Display.getDisplay(this);
    try
    {
        asker = new Kursorius();
        ats = asker.askServer("?v=d&t=1&pam_id="+ pamokuID[themes.getSelectedIndex()]);
        students = new List("Pasirinkite mokinį:",Choice.IMPLICIT);
        String temp="";
        int kiek=0;
        for (int i = 0; i < ats.length(); i++)
            {
                if (ats.charAt(i)=='|')
                    {
                        students.append(temp.substring(temp.indexOf("*")+1) , null);
                        mokiniuID[kiek] = temp.substring(0,temp.indexOf('*'));
                        temp = "";
                        kiek++;
                    }
                else {temp=temp+ats.charAt(i);}
            }
        students.addCommand(cmCont);
        students.addCommand(cmThemes);
        students.setCommandListener(this);
        display.setCurrent(students);
    }
}

```

```

    }
    catch (Exception e)
    { System.err.println("Msg2: " + e.toString());}
}

public void newMark(){ // Įrašome naują pažymį
display = Display.getDisplay(this);
tfMark = new TextField(students.getString(students.getSelectedIndex()), "", 2,
TextField.NUMERIC);
Form f = new Form("Irasyti ivertinima");
f.append(tfMark);
f.addCommand(cmSave);
f.addCommand(cmThemes);
f.setCommandListener(this);
display.setCurrent(f);
arNaujas = 1;
}

public void saveNewMark(){ //Saugome naują pažymį
try
{
asker = new Kursorius();
ats = asker.askServer("?v=n&t=1&pam_id="+ pamokuID[themes.getSelectedIndex()]+
"&p="+tfMark.getString()+"&mok_id="+mokinIUID[students.getSelectedIndex()]);
showMarks();
}
catch (Exception e)
{ System.err.println("Msg: " + e.toString());}
}

//Meniu pasirinkimo apdorojimas
public void commandAction(Command c, Displayable s)
{
if (c == cmReg) // Jei registravosi
{ mainForm();}
else if (c == cmExit) // Jei nori baigti darbą
{
destroyApp(false);
notifyDestroyed();
}
else if (c == cmMain) // Jei nori grįžti į pagrindinį meniu
{giveAction();}
else if (c == cmView) //Jei pasirinkta peržiūra
{
switch (kurEsam) // Tikriname, kuriame punkte buvo
{
case 0:
switch (actions.getSelectedIndex())
{
case 0:
wdList(); // Savaitės dienos pasirinkimas
break;
case 1:
groupList(); // Grupių sąrašas
}
break;
case 1:
groupListD();
break;
case 2:
showThemes();
break;
case 3:
showMarks();
}
}
}

```

```

        break;
    case 4:
        showMarks();
        break;
    case 5:
        showThemes();
        break;
    }
}
else if (c==cmWDays) // Reikia savaitės dienų sąrašo
{ wdList(); }
else if (c==cmNewM) // Reikia grupių sąrašo pagal data
{ selectStudent(); }
else if (c==cmCont) // Pasirinkus mokinį įrašome vertinimą
{ newMark(); }
else if (c==cmEditM) // Pažymio redagavimas
{ editMark(); }
else if (c==cmGList) // Grįžti prie grupių sąrašo
{ groupList(); }
else if (c==cmThemes) // Reikia pamokų temų sąrašo
{ showThemes(); }
else if (c==cmSave) // Saugoti pažimį
{
    if (arNaujas == 0)
        { saveEditedMark(); }
    else {saveNewMark();}
}
}

private void showAlert(String msg) // Pranešimas apie neįsisekimą
{
    alError = new Alert("Klaida", msg, null, AlertType.ERROR);
    alError.setTimeout(Alert.FOREVER);
    // Po klaidos parodymo, grįžtame į registracijos formą
    display.setCurrent(alError, fmMain);
}
}

```

Klasė kursorius.java

```
package com.eldien;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.InputStream;

import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;

/**
 * @author Eglė Bukienė
 */
public class kursorius {

    public static void main() {
    }
    public String askServer(String uzklausa) throws IOException
    {
        HttpConnection http = null;
        InputStream iStrm = null;
        boolean ret = false;
        // Formuojame užklausa
        String url = "http://62.80.248.212:9080/ed/edServlet" + uzklausa;
        try
        {
            http = (HttpConnection) Connector.open(url); //Nustatome jungtį
            http.setRequestMethod(HttpConnection.GET); //Nustatome jungimosi metoda GET
            iStrm = http.openInputStream(); //Atsitarome grįžtamąjį ryšį
            // Žiūrime, ką gavome
            String str;
            // Tikriname atsakymo statusą
            if (http.getResponseCode() == HttpConnection.HTTP_OK)
            {
                int length = (int) http.getLength();

                if (length != -1) // jei žinomas atsakymo ilgis
                {
                    byte servletData[] = new byte[length];
                    iStrm.read(servletData);
                    str = new String(servletData);
                }
                else // jei ilgis nežinomas...
                {
                    ByteArrayOutputStream bStrm = new ByteArrayOutputStream();
                    int ch;
                    while ((ch = iStrm.read()) != -1)
                        bStrm.write(ch);

                    str = new String(bStrm.toByteArray());
                    bStrm.close();
                }
            }
            // Perduodame atsakymą
            return str;
        }
        else
            // Perduodame servleto pasiųstą pranešimą apie klaidą
            str = "Klaida: " + new String( http.getResponseMessage());
            return str;
    }
}
```

```
finally
{ // Apsivalymas
  if (iStrm != null)
    iStrm.close();
  if (http != null)
    http.close();
}
}
```


Klasė SavDienos.java

```
package com.eldien;

import javax.microedition.lcdui.List;

/**
 * @author Eglė Bukienė
 */
public class SavDienos extends List{

    public SavDienos(String title, int listType) {
        super(title, listType);
        append("Pirmadienis", null);
        append("Antradienis", null);
        append("Trečiadienis", null);
        append("Ketvirtadienis", null);
        append("Penktadienis", null);
    }
}
```

10.3 3 priedas

Vartotojo vadovas mokiniui

Programą sudaro dvi bylos: MokEIDien.jad ir MokeIDien.jar. MokeIDien.jad byloje saugoma informacija apie programą, o MokeIDien.jar – sukompiliuota programa. Paleidus programą parodoma vartotojo registracijos forma:



Laukelyje „Vartotojo ID“ įrašykite sistemos administratoriaus suteiktą identifikavimo numerį, laukelyje „Slaptažodis“ – savo slaptažodį. Įrašę duomenis pasirinkite punktą „Options“ ir „Registruotis“. Įrašius neteisingus duomenis, bus parodytas pranešimas apie neteisingus duomenis



Paspaudę mygtuką „OK“ turėsite galimybę pakartoti registracijos procedūrą.
Teisingai įrašę prisijungimo duomenis, turite išvysti veiksmų pasirinkimo meniu:



Norėdami pamatyti pažymius, pasirinkite punktą „Pažymiai“ ir paspauskite „Options“ -> „Peržiūrėti“. Jums bus parodytas dalykų sąrašas, iš kurio turite pasirinkti, kurio dalyko pažymius norite pamatyti:



Pasirinkę norimą dalyką, paspauskite „Options“ -> „Peržiūrėti“ ir pamatysite pažymių sąrašą:



Norėdami peržiūrėti tvarkaraštį arba pamokų temas, pagrindiniame meniu pasirinkite punktą „Tvarkaraštis“ ir paspauskite „Options“ -> „Peržiūrėti“.



Pasirodžiusiame sąraše pasirinkite norimą savaitės dieną ir paspauskite „Options“ -> „Peržiūrėti“:



Pamatysite pasirinktos dienos pamokų tvarkaraštį:



Jei norite pamatyti kurio nors dalyko pamokų temas, pasirinkite ją tvarkaraštyje ir paspauskite „Options“ -> „Peržiūrėti“.



Pamatysite pasirinkto dalyko paskutiniųjų septynių pamokų temas:



Vartotojo vadovas mokytojui

Programą sudaro dvi bylos: MktEIDien.jad ir MktEIDien.jar. MktEIDien.jad byloje saugoma informacija apie programą, o MktEIDien.jar – sukompiliuota programa. Paleidus programą parodoma vartotojo registracijos forma:



Laukelyje „Vartotojo ID“ įrašykite sistemos administratoriaus suteiktą identifikavimo numerį, laukelyje „Slaptažodis“ – savo slaptažodį. Įrašę duomenis pasirinkite punktą „Options“ ir „Registruotis“. Įrašius neteisingus duomenis, bus parodytas pranešimas apie neteisingus duomenis. Paspaudę mygtuką „OK“ turėsite galimybę pakartoti registracijos procedūrą.

Teisingai įrašę prisijungimo duomenis, turite išvysti veiksmų pasirinkimo meniu:



Norėdami peržiūrėti tvarkaraštį, pagrindiniame meniu pasirinkite punktą „Tvardarastis“ ir paspauskite „Options“ -> „Peržiūrėti“.



Pasirodžiusiame sąrašė pasirinkite norimą savaitės dieną ir paspauskite „Options“ -> „Peržiūrėti“:



Pamatysite pasirinktos dienos grupių sąrašą:



Jei norite pamatyti kurio nors dalyko pamokų temas, pasirinkite jį tvarkaraštyje ir paspauskite „Options“ -> „Peržiūrėti“. Pamatysite pasirinkto dalyko paskutiniųjų septynių pamokų temas:



Pasirinkę norimą temą ir paspaudę „Options“ -> „Peržiūrėti“, išvysite tos pamokos mokinių vertinimus.



Norėdami keisti arba įrašyti kitą įvertinimą paspauskite „Options“ ir pasirinkite atitinkamą punktą – „Keisi“ arba „Įrašyti nauja“. Tuomet atsiradusiame laukelyje (naujo pažymio įrašymo atveju prieš tai dar reikės pasirinkti mokinį, kuriam įrašomas vertinimas) įrašykite pažymį ir spauskite „Saugoti“.

