

LATVIJAS UNIVERSITĀTE  
DATORIKAS FAKULTĀTE

**MOBILO IEKĀRTU PROGRAMMATŪRAS IZSTRĀDES  
IESPĒJU SALĪDZINĀŠANA DAŽĀDĀS  
OPERĒTĀJSISTĒMĀS.**

BAKALaura DARBS

Autors: **Oļegs Nečiporuks**

Stud. apl. DatZ030110

Darba vadītājs: lektors M.sc.comp. Uldis Straujums

RĪGA 2009

## **Anotācija**

Mūsdienās mobilie telefoni vairs nav vienkārši telefoni. Tos pat jau ir grūti nosaukt vienkārši par telefonu, jo zvanīšanas funkcija ir tikai viena no daudzajām mobilo iekārtu funkcijām. Lai mobilā iekārta spētu paveikt visas šīs visdažādākās funkcijas, tai ir nepieciešama sarežģīta operētājsistēma un programmatūra.

Bakalaura darbā tiek salīdzinātas programmatūras izstrādes iespējas populārākajām operētājsistēmām. Visu darbu var sadalīt četrās daļās.

Pirmajās trīs daļās ir apskatītas tirgū populārākās mobilo iekārtu operētājsistēmas, izpētītas iespējas programmatūras izstrādāšanai šīm operētājsistēmām.

Ceturtajā daļā tiek salīdzinātas programmatūras izstrādes iespējas apskatītajām operētājsistēmām.

## **Abstract**

In modern life mobile phones are something sophisticated that differ from ordinary phones. Sometimes it is even hard to call them phones, because a call function is not anymore the only function modern mobile phones have. To support a variety of functions and facilities mobile devices come with sophisticated operating systems and software.

This paper gives a comparison of software development for the most popular operational systems for mobile devices. The paper is divided into four main parts.

In the first three parts is given an overview of the most popular operational systems for mobile devices in the market, and made a research of software development possibility for these systems.

The fourth part gives a comparison of software development possibilities for these operational systems.

## **Atslēgvārdi**

Windows Mobile, Symbian OS, iPhone OS, programmatūras izstrāde.

# SATURS

Ievads.....	6
1. Programmatūras izstrādes iespējas priekš mobilajām iekārtām ar Windows Mobile operētājsistēmu .....	7
1.1. Operētājsistēmas apraksts.....	8
1.2. Programmatūras izstrādes iespējas, izmantojot operētājsistēmas izstrādātāja sniegtos rīkus	12
1.3. Programmatūras uzstādīšana mobilajā iekārtā .....	17
2. Programmatūras izstrādes iespējas priekš mobilajām iekārtām ar Symbian OS operētājsistēmu .....	19
2.1. Operētājsistēmas apraksts.....	20
2.2. Programmatūras izstrādes iespējas, izmantojot operētājsistēmas izstrādātāja sniegtos rīkus	24
2.3. Programmatūras uzstādīšana mobilajā iekārtā .....	27
3. Programmatūras izstrādes iespējas priekš mobilajām iekārtām ar iPhone OS operētājsistēmu	29
3.1. Operētājsistēmas apraksts.....	30
3.2. Programmatūras izstrādes iespējas, izmantojot operētājsistēmas izstrādātāja sniegtos rīkus	33
3.3. Programmatūras uzstādīšana mobilajā iekārtā .....	36
4. Salīdzināšana .....	37
Secinājumi .....	41
Nobeigums.....	42
Izmantotā literatūra un avoti.....	43
Pielikumi.....	44
1. pielikums. .pkg faila formāts .....	44

## Ievads

Mūsdienās mobilās iekārtas tiek izmantotas aizvien vairāk un vairāk. Mobilo iekārtu tirgū ir ļoti liels produktu klāsts, un visas šīs mobilās iekārtas izmanto dažādas operētājsistēmas.

Bakalaura darbā tiek izpētīts programmatūras izstrādes iespējas priekš populārākajām operētājsistēmām, izmantojot operētājsistēmu sniegtos programmu izstrādes rīkus, kā arī tiek aprakstīts primitīvu programmu izveidošanas procesi priekš aprakstītām operētājsistēmām un veidi, kā var ielādēt un uzinstalēt izveidoto programmu mobilajā iekārtā.

Darbā ir izvēlēts aprakstīt 3 populārākās operētājsistēmas. Sākumā tiek apskatīta *Microsoft* kompānijas *Windows Mobile* operētājsistēma un programmatūras izstrādes iespējas tajā. Pēc tam tiek apskatīta *Nokia* kompānijas operētājsistēma *Symbian* un lietojumu izstrādes iespējas šai operētājsistēmai. Tālāk tiek apskatīta *Apple* korporācijas *iPhone* telefona operētājsistēma un paņēmieni, kā var izstrādāt programmas priekš šīs operētājsistēmas. Tālāk, kad visas 3 operētājsistēmas un programmatūras izstrādes iespējas tajās ir apskatītas, tās tiek salīdzinātas.

## **1. Programmatūras izstrādes iespējas priekš mobilajām iekārtām ar Windows Mobile operētājsistēmu**

Viena no mūsdienās populārākajām mobilo iekārtu operētājsistēmām ir *Windows Mobile*. Operētājsistēma *Windows Mobile* ir kompānijas *Microsoft* izstrādājums. Šī operētājsistēma tiek izmantota daudzu mobilo iekārtu izstrādātāju dažādās ierīcēs. Tā var strādāt gan ierīcēs ar skārienjūtīgu ekrānu, gan arī bez tā. Mobilā iekārta, uz kuras tiek darbināta *Windows Mobile* operētājsistēma var būt gan ar GSM-moduli, gan bez tā. To izmanto tādi mobilo iekārtu ražotāji kā: *Samsung, HTC, LG, Sony Ericsson, HP*, u.c.

## 1.1. Operētājsistēmas apraksts

Nedaudz ieskatoties vēsturē, kompānija *Microsoft* ir izlaidusi šādas mobilo iekārtu operētājsistēmas (4):

- Pocket PC 2000 (Windows CE 3.0 Pocket PC Edition);
- Pocket PC 2002;
- Windows Mobile 2003 (Pocket PC 2003);
- Windows Mobile 2003 SE (Pocket PC 2003 Second Edition);
- Windows Mobile 5.0;
- Windows Mobile 6;
- Windows Mobile 6.1;
- Windows Mobile 6.5.

*Windows Mobile 6.5* tika izlaists 2009. gada 1. martā. Darba rakstīšanas laikā šī ir pēdējā *Windows Mobile* versija, bet tā kā oficiālajā mājas lapā vēl nebija nekādas informācijas par programmatūras izstrādes iespējām, tad darbā tiek apskatīta *Windows Mobile 6.1* versija (tālāk tiks saukta kā *WM6*).

*WM6* ir veidota uz *Windows CE 5.2* bāzes. *Windows Mobile 6.1* operētājsistēma iedalās trijās kategorijās:

- Classic;
- Professional;
- Standard.

*WM6 Classic* tiek lietota mobilajās iekārtās, kurās ir skārienjūtīgs ekrāns, bet nav GSM-moduļa. *Professional* versija tiek lietota mobilajās iekārtās, kurām ir gan skārienjūtīgs ekrāns, gan GSM-modulis. Un *Standard* operētājsistēmas versija tiek lietota mobilajās iekārtās, kurās nav skārienjūtīga ekrāna, bet ir GSM-modulis. Atšķirības starp *Classic* un *Professional WM6* operētājsistēmām ir nelielas. Vienīgā atšķirība ir tāda, ka *Professional* versijai ir pielikta klāt visa, zvaniem nepieciešamā, programmatūra. Toties *Standard* versija ievērojami atšķiras no *Professional* un *Classic* dēļ tā, ka tai nav atbalsta skārienjūtīgam ekrānam. Programmas, kuras ir domātas *Standard* versijai nevar uzstādīt uz *Professional* un *Classic* versijām, kā arī otrādi (1).

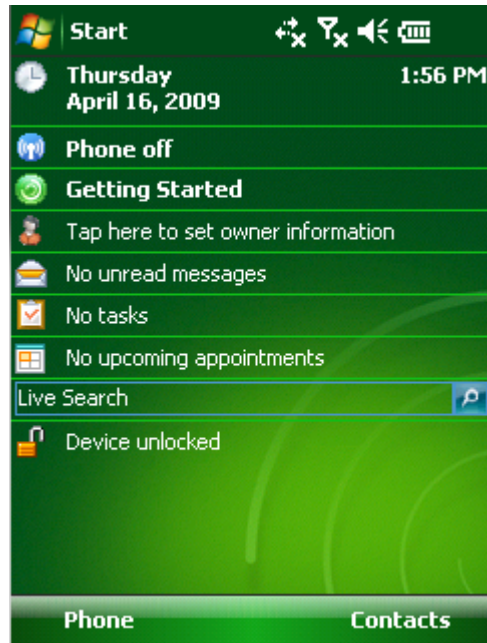
Lai operētājsistēma varētu strādāt, mobilajai iekārtai jāatbilst šādām minimālajām prasībām:

- Flash ROM – 32 Mb;
- Operatīvā atmiņa – 32 Mb;

- Procesora frekvence – 200MHz.





Apskatīsim iespējas, ko mobilās iekārtas lietotājam sniedz pati operētājsistēma. Pirmo reizi ieslēdzot aparātu, operētājsistēma prasa uzstādīt paroli iekārtai. Ir iespēja ievadīt kaut kādu tekstu, kas palīdzēs atcerēties paroli gadījumā, ja lietotājs ir aizmirsis to.


Operētājsistēmas interfeiss ir redzams attēlā 1.1.

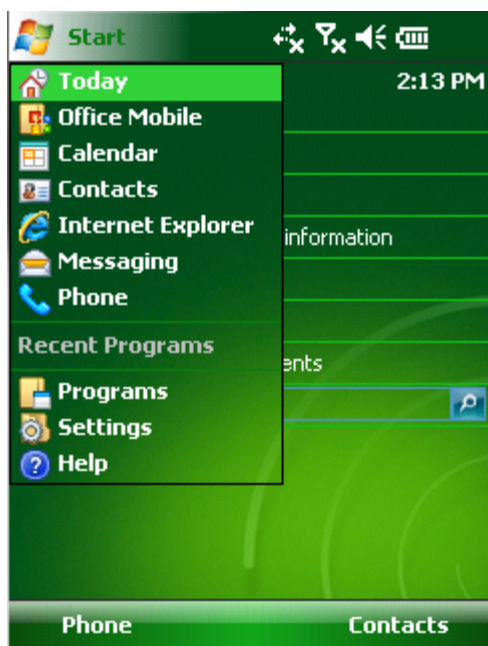


1.1. att. WM6 operētājsistēma sākuma ekrāns


Augšējā joslā ir 4 indikatori:

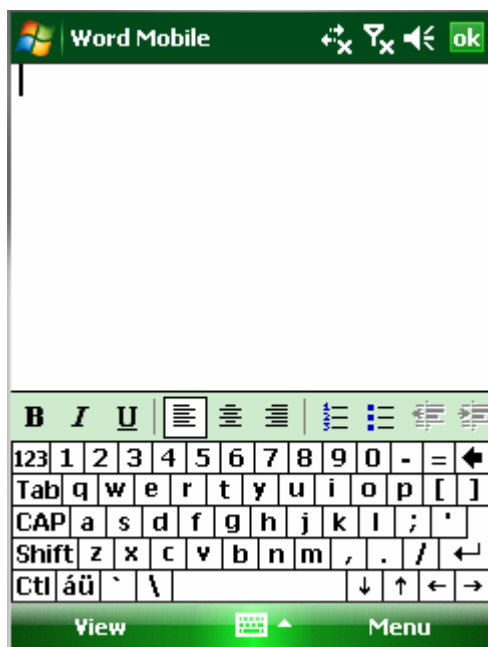
-  - WiFi tīkla statusa indikators;
-  - GSM tīkla statusa indikators;
-  - skaļuma statusa indikators;
-  - baterijas statusa indikators.

Augšējā kreisajā stūrī ir poga „Start” , nospiežot kuru, atveras galvenā izvēlne. Izvēlnes izskats ir redzams attēlā 1.2.



1.2. att. WM6 operētājsistēmas galvenās izvēlnes izskats

Programmās, kurās ir nepieciešams rakstīt, ir pieejama QWERTY klaviatūra. Loga apakšā ir speciāla poga , nospiežot kuru var parādīt vai noslēpt klaviatūru. Klaviatūra ir redzama attēlā 1.3.



1.3. att. WM6 operētājsistēmas klaviatūra

WM6 operētājsistēmā ir daudz dažādu programmu, kuras ir uzstādītas operētājsistēmā pēc noklusējuma:

- Office Mobile aplikācijas, tādas kā Word Mobile, Excel Mobile un PowerPoint Mobile;
- Kalendārs ar iespēju veidot piezīmes konkrētos datumos un laikos;

- Kontaktu grāmata;
- Interneta vietņu pārlūks Internet Explorer;
- E-pasta un īsziņu sūtīšanas un lasīšanas;
- Telefona programma, kura ļauj mobilo iekārtu, kurā ir iebūvēts GSM modulis, izmantot kā telefonu;
- Dažādas sistēmas uzstādījumu mainīšanas programmas, tādas kā klaviatūras konfigurācijas, mobilas iekārtas paroles uzstādījumi vai skaņas uzstādījumu rediģēšanas programma;
- Operētājsistēmas lietošanas palīgs, kas lietotājiem ar nepietiekošām operētājsistēmas zināšanām uzzināt par to, kā tiek veikta kāda noteikta darbība (piemēram, kā var nosūtīt vēstuli uz e-pastu);
- Spēles;
- ActiveSync – programma paredzēta iekārtas sinhronizācijai ar datoru;
- Kalkulators;
- Datņu pārlūks;
- Programma koplietojamu datņu veidošanai;
- Windows Live Messenger – sociālā tīkla programma;
- Notes – programma, kas ļauj veidot piezīmes;
- Bilžu pārlūkprogramma;
- Meklētājs, kas ļauj meklēt pēc atslēgas vārdiem dažādās vietās (piemēram: kontaktu sarakstā vai īsziņās);
- Uzdevumu menedžeris, kas ļauj apskatīt operētājsistēma izpildāmos procesus;
- Uzdevumu plānotājs;
- Meklētājs, kas ļauj meklēt pēc atslēgas vārdiem internetā;
- Windows Media – programma, kas ļauj atskaņot audio failus un skatīties video failus.

## 1.2. Programmatūras izstrādes iespējas, izmantojot operētājsistēmas izstrādātāja sniegtos rīkus

Kompānija *Microsoft* ir izstrādājusi programmatūras izstrādātājiem, kas vēlas veidot lietojumus priekš mobilajām iekārtām, kas strādā ar *WM6* operētājsistēmu. Šī rīku kopa iekļauj sevī dokumentāciju, kodu piemērus, galvenes, bibliotēkas un emulatorus, kuri ir kā papildinājums *Visual Studio* programmēšanas videi. Bakalaura darba rakstīšanas brīdī pēdējā šī rīku kopa bija *Windows Mobile 6 SDK Refresh*. *Windows Mobile 6 SDK Refresh* ir publicēta 2007. gada janvārī (3).

Tātad lai varētu izstrādāt programmatūru priekš mobilajām iekārtām, kas strādā ar *WM6* operētājsistēmu ir nepieciešama programmēšanas vide *Visual Studio*, *Windows Mobile 6 SDK* un *ActiveSync* vai *Windows Mobile Device Center*. *Windows Mobile 6 SDK* ir iespējams lejupielādēt no *Microsoft* oficiālās mājas lapas internetā un tā ir pieejama bezmaksas. *Visual Studio* arī var lejupielādēt no *Microsoft* oficiālās mājas lapas internetā, bet tai ir vairāki iespējami varianti:

- *Visual Studio 2008 Professional Edition Trial*. Šī ir bezmaksas versija, kuru varēs lietot 90 dienas no instalācijas brīža. Tai ir pilna funkcionalitāte.

Minimālās sistēmas prasības programmas uzstādīšanai ir:

- Centrālā procesora jauda vismaz 1.6 Ghz;
- 384 MB operatīvā atmiņa (768 MB operatīvā atmiņa, ja ir *Windows Vista*);
- 2.2 GB brīvas vietas uz cietā diska;
- Cietais disks ar ātrumu vismaz 5400 RPM;
- Ekrāna izšķirtspēja vismaz 1024 x 768 pikseļi;
- DVD-ROM dzinis;
- *Visual Studio Team System 2008 Development Edition with MSDN Premium*. Šī ir maksas versija. Tā maksā 5469 USD un to ir iespējams nopirkt *Microsoft* oficiālajā mājas lapā internetā. Minimālās sistēmas prasības ir tādas pašas, kā *Visual Studio 2008 Professional Edition Trial* versijai (arī turpmākām versijām šajā sarakstā būs tādas pašas minimālās sistēmas prasības, tāpēc turpmāk tās netiks aprakstītas). Šī versija ir domāta izstrādātāju komandām un tajā ir iekļauti rīki, kas palīdzēs programmu kodu veidot kvalitatīvāku, drošāku un palīdz izsargāties no kļūdām turpinot projektu izstrādi.
- *Visual Studio 2008 Professional Edition*. Šī versija pēc iespējām ir tāda pati, kā

bezmaksas versija, tikai tai nav ierobežojumu lietošanas laikā un tā ir maksas versija. To ir iespējams nopirkt Microsoft oficiālajā mājas lapā internetā un tā maksā 799 USD.

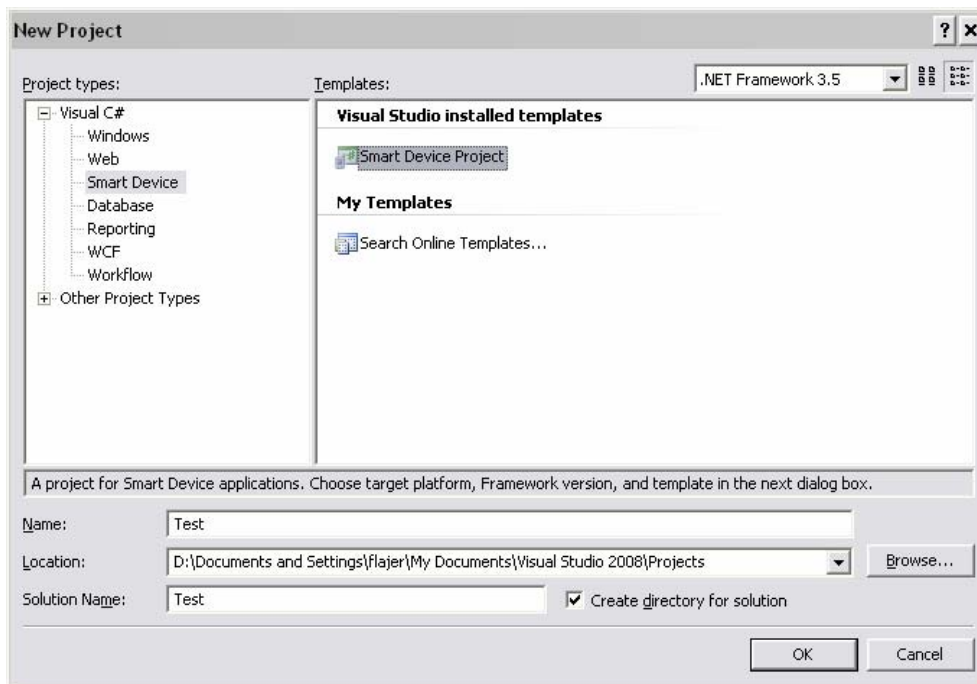
- *Visual Studio Team System 2008 Team Suit with MSDN Premium*. Šajā versijā ietilpst pilnīgs iespēju klāsts. Ar to var strādāt jebkurš no izstrādātāju komandas, gan arhitekts, gan dizaineris, gan izstrādātājs, gan datubāzes administrators, gan arī testētājs. Tā maksā 10939 USD.
- Bez šīm versijām ir vēl citas versijas, bet tās ir domātas kādam noteiktam izstrādāšanas posmam, piemēram, atsevišķa versija darbam ar datubāzi, vai atsevišķa versija testēšanai. Tādas versijas darbā netiks aprakstītas.

*ActiveSync* vai *Windows Mobile Device Center* programmas ir nepieciešamas, lai izveidoto programmu varētu uzstādīt mobilajā iekārtā vai mobilās iekārtas emulatorā. Operētājsistēmā *Windows Vista* jau ir iebūvēta programma *Windows Mobile Device Center*, tāpēc papildus neko instalēt nav nepieciešams. Ja uz datora tiek izmantota *Windows XP* operētājsistēma, tad ir nepieciešams uzstādīt *ActiveSync* programmu, kura bezmaksas ir pieejama *Microsoft* mājas lapā internetā (2). Programmas tiek rakstītas izmantojot *C#* programmēšanas valodu, tāpēc lai varētu rakstīt programmas priekš mobilajām iekārtām ar *WM6* operētājsistēmu, ir arī jāzina *C#* programmēšanas valoda.

*Microsoft* mājas lapā internetā ir pieejams ļoti daudz dažādas pamācošas informācijas par to, kā var veidot programmatūru priekš *WM6* operētājsistēmas. Informācija ir pieejama gan teksta veidā angļu valodā, gan arī video formātā video apmācību veidā. Mājas lapā viegli saprotamā veidā ir pastāstīts par to, kas ir nepieciešams programmētājam, lai varētu izveidot savu programmu priekš *WM6* operētājsistēmas un kas ir jādara, lai to varētu izdarīt.

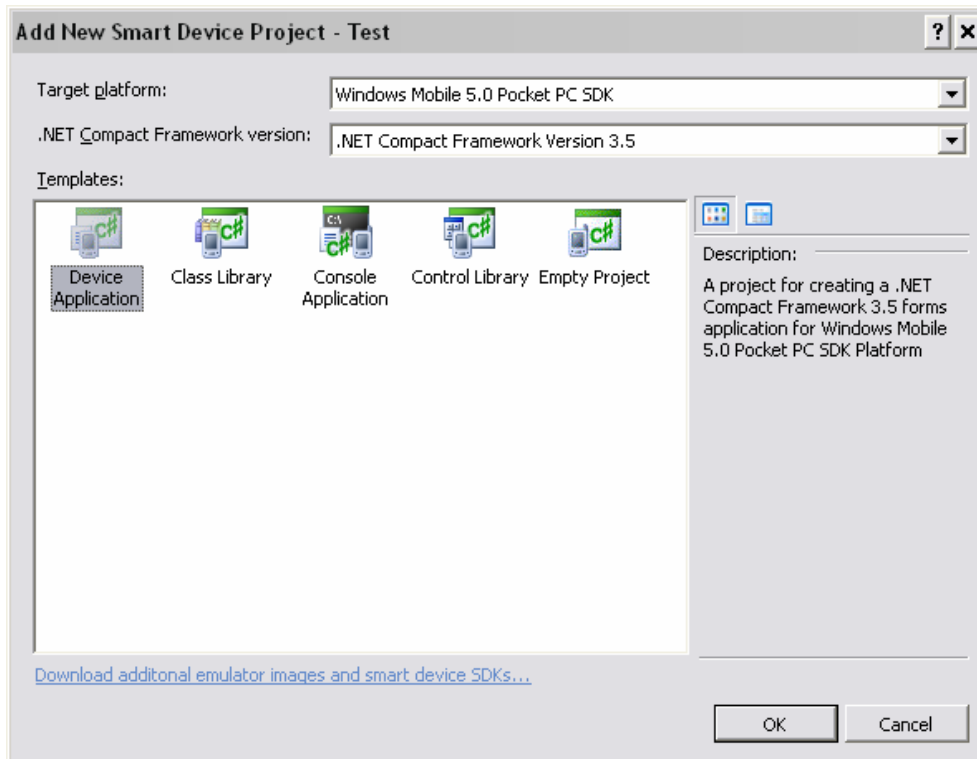
Lai varētu izveidot triviālu programmu priekš *WM6* operētājsistēmas ir nepieciešams:

- *Izveidot jaunu mobilās iekārtas projektu*. Nepieciešams atvērt *Visual Studio* programmēšanas vidi. Tad **File** izvēlnē izvēlēties **New > Project**. Laukā ar projektu tipiem atveram sadaļu **Visual C#** un izvēlamies **Smart Device**. Teksta laukā **Name** var ierakstīt jaunā projekta nosaukumu, bet var atstāt to tādu, kāds tas ir. Tad spiežam pogu **OK**. Attēlā 1.4. parādīts jauna projekta tipa izvēles logs.



1.4. att. Jauna projekta tipa izvēles logs

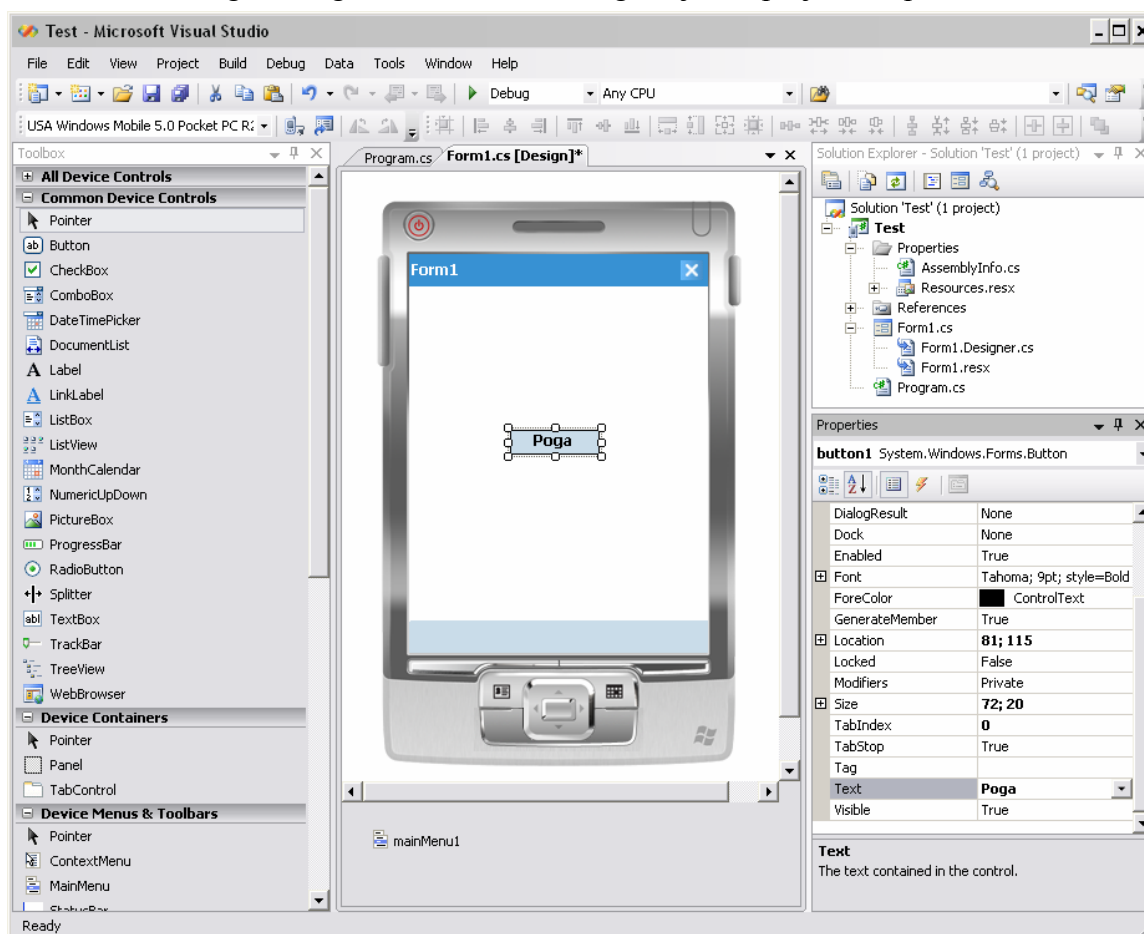
Logā **Add New Smart Device Project** ir jāizvēlas **Device Application** un jānospiež poga **OK**.  
 Logs **Add New Smart Device Project** ir parādīts attēlā 1.5.



1.5. att. Add New Smart Device Project logs

- Pievieno jaunam projektam funkcionālu. Kreisajā malā rīku kastē (**Toolbox**) izvēlamies rīku ar nosaukumu **Button** un ar peles kursoru pārnesam to uz mobilās ierīces ekrāna.

Labajā malā apakšā īpašību kastē (**Properties**) pogai nomainām tās īpašības **Text** vērtību uz tekstu „Poga”. Programmēšanas vides logs ar jaunu projektu ir parādīts attēlā 1.6.



1.6. att. Programmēšanas vides logs

Divas reizes nospiežam un jaunizveidotās pogas, lai izveidotu notikuma apdarinātāju.

Apdarinātājā ierakstam sekojošu kod fragmentu:

```
button1.Text = "Poga strādā!";
```

- *Nokompilēt un pārbaudīt darbību.* Spiežam uz klaviatūras taustiņu F5, vai programmēšanas vidēs *Visual Studio* rīku joslā spiežam pogu **Start Debugging** . Pēc veiksmīgas kompilēšanas programma pajautās, kur Jūs vēlaties izvietot savu jauno programmu. Izvēlamies vienu no emulatoriem. Pēc brīža atvērsies mobilās iekārtas emulatora logs, kuram ekrāna centrā būs poga ar tekstu „Poga”. Nospiežot šo pogu, tās teksts nomainīsies uz tekstu „Poga strādā!”.

Triviālas Programmas piemēra veidošanai tika izmantotas šādas programmatūras:

- Visual Studio 2008 Professional Edition Trial;
- Windows Mobile 6 SDK Refresh;
- ActiveSync.

Piemērs tika paņemts no kompānijas *Microsoft* mājas lapas internetā (2).

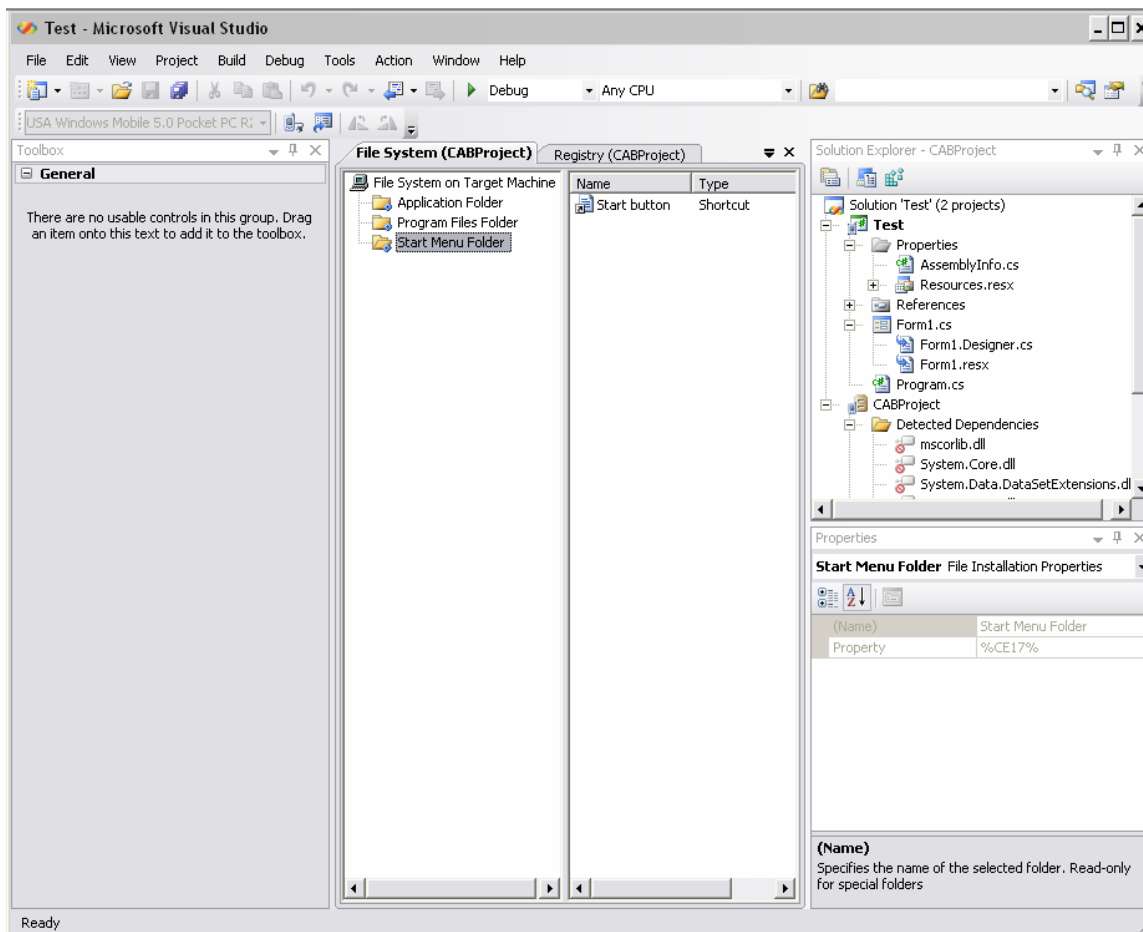
Programmas veidošana notiek samērā vienkārši un ātri. Lai pārbaudītu, vai izveidotā programma darbojas korekti, nav nepieciešam mobilā ierīce. To var aizstāt ar mobilās iekārtas emulatoru, kas ir ļoti ērti tāpēc, ka pateicoties mobilās iekārtas emulatoram, programmatūras veidotājam var arī nebūt pati mobilā iekārt.

### 1.3. Programmatūras uzstādīšana mobilajā iekārtā

Lai izveidoto programmu uzstādītu mobilajā iekārtā ar *WM6* operētājsistēmu, ir nepieciešams šai programmai izveidot *CAB* failu. To var izdarīt šādi:

- Programmēšanas vidē *Visual Studio* atveram projektu, kuram gribam izveidot instalācijas failu priekš mobilās iekārtas un pārlicināmies, ka tas kompilējas un strādā korekti.
- Pārlicināmies, ka projekta pārlūks (**Solution Explorer**) ir aktivizēts. Ja tas tā nav, tad **View** izvēlnē izvēlamies **Solution Explorer**.
- **File** izvēlnē izvēlamies **Add > New Project**. Atvēršies **Add New Project** logs.
- **Project Type** panelī atveram **Other Project Type** un izvēlamies **Setup and Deployment**. Pēc tam **Template** panelī izvēlamies **Smart Device CAB Project**. Laukā **Name** var ierakstīt projekta nosaukumu, bet var arī atstāt to tādu, kāds tas ir.
- Īpašību kastē var uzstādīt projekta nosaukumu, nomainot īpašību **ProductName**. Ar šādu nosaukumu programma būs redzama instalācijas mapē, kā arī **Add or Remove Programs** dialogā (piemēram, dzēšot programmu).
- Labajā panelī **File System Editor** dialogā klikšķinām uz **Application Folder**, lai uzstādītu, ka instalējot programmu, tā tiks uzinstalēta šajā mapē. Ja jūs neredzat **File System Editor** dialogu, tad spiediet ar labo peles taustiņu uz *CAB* projekta nosaukuma projekta pārlūkā un izvēlnē izvēlaties **View -> File System**.
- **Action** izvēlnē izvēlās **Add > Project Output**. Atveras **Add Project Output Group** logs.
- Šajā logā **Project** izvēlnē izvēlamies mūsu projektu, izvēlamies **Primary output** un spiežam pogu **OK**.
- Ja nepieciešams izveidot īsinājumikonu **Start** izvēlnē vai **Programs** mapē, tad:
  - Labajā panelī **File System Editor** dialogā klikšķinām uz **Application Folder** un tad uz **Primary Output from <jūsu projekta nosaukums>**.
  - Izvēlnē **Action** izvēlamies **Create Shortcut to Primary Output from <jūsu projekta nosaukums>**. Uzklikšķinām uz īsinājumikonas ar labo peles taustiņu, izvēlamies **Rename** un pārsaucam īsinājumikonu tā kā vēlamies.
  - Labajā panelī **File System Editor** dialogā klikšķinām uz laukuma ar labo peles taustiņu un izvēlnē izvēlamies **Add Special Folder > Start Menu Folder** vai **Programs folders**, atkarībā no tā, kur tieši Jūs vēlaties izveidot īsinājumikonu

savai programmai. Pēc tam atveram mapi **Application Folder** un ar peles kursoru pārnesam īsinājumikonu uz jaunizveidoto mapi. Attēlā 1.7. ir redzams *Visual Studio* programmēšanas vides logs ar izveidotu **Start Menu Folder** direktoriju un ievietotu tajā īsinājumikonu.



1.7. att. Programmēšanas vides logs

- Izvēlnē **Build** izvēlamies **Build CABProject**, pēc tam izvēlnē **File** spiežam **Save All**.
- Direktorijā, kurā tiek saglabāts Jūsu projekts apakšdirektorijā **CABProject\Debug** atrodas jaunizveidotais **CAB** fails, kuru ar **ActiveSync** programmas palīdzību var pārkopēt uz mobilās iekārtas. Tad kad fails ir pārkopēts mobilajā iekārtā, tad atliek tikai izmantojot pašu mobilo iekārtu palaist šo failu un **WM6** operētājsistēma pati uzinstalēs šo programmu (2).

## 2. Programmatūras izstrādes iespējas priekš mobilajām iekārtām ar Symbian OS operētājsistēmu

Otra ļoti populāra mobilo iekārtu operētājsistēma ir *Symbian OS*. Šo operētājsistēmu izstrādā *Symbian* konsorcijs, kurš ir dibināts 1998 gadā. To dibināja četras kompānijas: *Psion*, *Nokia*, *Ericsson*, *Motorola*. Vēlāk šim konsorcijam pievienojās arī tādas kompānijas, kā *Sony Ericsson*, *Siemens*, *Panasonic*, *Fujitsu*, *Samsung*, *Sony* un *Sanyo* (7).

2008. gada 24. jūnijā kompānijas *Nokia*, *Sony Ericsson*, *Motorola* un *NTT DOCOMO* oficiāli paziņoja par to, ka apvienos *Symbian OS*, *S60*, *UIQ* un *MOAP(S)* operētājsistēmas ar mērķi izveidot vienotu atvērtu mobilo platformu. Kopā ar kompānijām *AT&T*, *LG Electronics*, *Samsung Electronics*, *STMicroelectronics*, *Texas Instruments* un *Vodafone* tika izveidota nekomerciāla organizācija *Symbian Foundation* (7).

Jau vairākus gadus *Symbian OS* ir populārākā mobilo iekārtu operētājsistēma. 2005. gadā tā tika izmantota 59,7% no visiem pasaules viedajiem telefoniem. 2006. gadā *Symbian OS* daļa jau bija 72,8%. Pēc tam ar katru gadu šis skaitlis sāka pamazām krist, bet operētājsistēma aizvien paliek pirmajā vietā. 2007. gadā šī operētājsistēma tika izmantota 65,5% viedos telefonos un 2008. gadā jau tikai 57.1%.

## 2.1. Operētājsistēmas apraksts

*Symbian OS* ir izveidota no *EPOC32* sistēmas, kuru izveidoja kompānija *Psion*. 1998. – 2000. gadā lielākā daļa sistēmas tika pārrakstīta ar mērķi optimizēt kodu, tā lai sistēma varētu strādāt uz ierīcēm ar ierobežotiem resursiem. Izstrādātājiem izdevās panākt ievērojamu atmiņas ekonomiju, uzlabot koda kešošanu, kā rezultātā uzlabojās programmu ātrdarbība un samazinājās elektroenerģijas patēriņš. No izstrādes viedokļa, atšķirīgā sistēmas īpašība ir tas, ka operētājsistēmas arhitektūra ir pilnība objektorientēta. Sākot ar operētājsistēmas 9. versiju parādījās nopietns aizsardzības mehānisms – *API* atdalīšana saskaņā ar programmu tiesībām. Galvenā valoda, kurā tiek veidotas programmas priekš *Symbian OS* ir *C++* programmēšanas valoda, kā arī ir *Java* programmēšanas valodas atbalsts (7).

Nedaudz ieskatoties vēsturē:

- 1999. gadā kompānija *Psion* izlaida operētājsistēmu *EPOC Release 5*, kuras otrs nosaukums bija *Symbian OS v5*.
- 2000. gadā parādījās *EPOC5u* sistēma (*Symbian OS v5.1*).
- 2001. gadā iznāk *Symbian OS v6.0* un pēc tam arī *Symbian OS v6.1*. Pirmais viedais telefons, kurš izmantoja šo operētājsistēmu bija *Nokia 9210*.
- 2003. gadā nāk klajā *Symbian OS v7.0* un *v7.0s*. Šī sistēma atbalstīja *UIQ*, *Series 80*, *Series 60* un *FOMA* platformas.
- 2004. gadā parādās *Symbian OS v8.0* un *v8.1*, kuru īpatnība bija iespēja izvēlēties starp diviem dažādiem kodoliem (*EKA1* un *EKA2*).
- 2004. gadā vēl parādās operētājsistēma *Symbian OS v9.0*. Sākot ar šo versiju pārstāja izmantot *EKA1* kodolu. Bet pirmais telefons, kurš izmantoja šo operētājsistēmu, parādījās tikai 2006. gadā.
- 2005. gada sākumā parādījās *Symbian OS v9.1*.
- 2006. gada pirmajā kvartālā parādījās *Symbian OS v9.2* ar *Bluetooth 2.0* un *OMA Device Management 1.2* atbalstu.
- 2006. gada 12. jūlijā oficiāli tika izlaista operētājsistēma *Symbian OS v9.3*.
- 2007. gada martā parādījās *Symbian OS v9.5*. Ievērojamākie jauninājumi šajā operētājsistēmā bija enerģijas patēriņa samazināšana par 20% – 30%, līdz 75% samazināts programmu ielādes ātrums, uzlabots *VoIP* atbalsts, kā arī ciparu televīzijas

atbalsts.

Eksistē vairākas *Symbian OS* modifikācijas (dažādiem mobilo iekārtu tipiem). Pašas populāras no tām ir:



- **UIQ** ir galvenā operētājsistēma priekš viedajiem telefoniem, kurus ražo kompānijas *Sony Ericsson* un *Motorola*. Atšķirīgā šīs operētājsistēmas īpatnība ir tāda, ka tā atbalsta mobilās ierīces ar sensoru ekrānu. 2008. gadā šo platformu pārstāja izstrādāt.
- **Series 60** ir galvenā operētājsistēma priekš *Nokia* kompānijas viedajiem telefoniem. Izstrādāta ierīcēm ar telefona klaviatūru.
- **Series 80** arī ir *Nokia* kompānijas izstrādājums viedajiem telefoniem ar pilnizmēra klaviatūru.
- **FOMA** (*Freedom of Mobile Access*). Šī platforma tiek izplatīta Japānā. Tā tiek bīdīta ar *NTT DoCoMo* mobilo sakaru kompānijas palīdzību. Ierīces šai platformai ražo kompānijas *Mitsubishi*, *Fujitsu*, *Sony Ericsson* un *Sharp*.

Bakalaura darba izstrādes brīdī jaunākā *Symbian OS* versija bija *v9.5*, bet tā kā oficiālā izstrādes rīku kopa šai operētājsistēmas versijai bakalaura izstrādes brīdī nebija pieejama, tad darbā tiks apskatīta *Symbian OS v9.4* priekš *Nokia N97* viedā telefona. Tā ir jaunākā versija, kurai ir pieejama oficiālā izstrādes rīku kopa.

Apskatīsim pašu operētājsistēmu. Pirmo reizi ieslēdzot mobilo ierīci, operētājsistēma paprasa ievadīt datumu un laiku. *Symbian OS v9.4* interfeiss ir redzams attēlā 2.1.



2.1. att. *Symbian OS v9.4* sākuma ekrāns

Ekrāna augšā ir GSM tīkla indikators  un akumulatora enerģijas daudzuma indikators . Vēl

operētājsistēmas sākuma ekrānā šādi widgeti:

- Pulkstenis, kurš rāda sistēmas laiku;
- Datums;
- Izvēlēta profila nosaukums;
- Mūzikas atskaņotais, kurš attēlo informāciju par pašlaik spēlēto dziesmu (ja mūzikas atskaņotais ir ieslēgts);
- Widgets a 4 biežāk izmantojamajām programmām;
- Plānotāja widgets, kurš parāda aktuālos notikumus no plānotāja.

Operētājsistēmā pēc noklusējuma ir uzstādītas vairākas programmas. Galvenās izvēlnes ekrāns ir redzams attēlā 2.2.



2.2. att. Symbian OS v9.4 galvenās izvēlnes ekrāns

Operētājsistēmā *Symbian OS v9.4* ir daudz dažādu programmu, kuras ir uzstādītas sistēmā no operētājsistēmas instalācijas brīža:

- Kalendārs un plānotājs;
- Kontaktu grāmata;
- Mūzikas atskaņotais;
- Interneta pārlūks;
- Īsziņu un e-pastu sūtīšanas un saņemšanas programma;
- Attēlu pārlūks;
- Video un TV pārraižu atskaņotais;
- Sistēmas konfigurēšanas rīki;

- Kalkulators;
- Pulkstenis ar modinātāju;
- Vienību konverters, kurš var pārkonvertēt dažādas valūtas, mērvienības u.c.;
- Piezīmju blociņš;
- Skaņas ierakstītājs (diktofons);
- Foto un Video kameras vadīšanas programma;
- Dažādas programmas atrašanās vietas noteikšanai, tādas kā kompass, karte;
- Failu pārlūks;
- Vārdnīca;
- Konsole.

## 2.2. Programmatūras izstrādes iespējas, izmantojot operētājsistēmas izstrādātāja sniegtos rīkus

Tā kā operētājsistēma *Symbian OS* nav vienas kompānijas izstrādājums, bet gan daudzu kompāniju un tā kā šai operētājsistēmai programmas var rakstīt dažādās programmēšanas valodās, tad rīku programmu izstrādei ir daudz un dažādas. Bakalaura darbā tiks apskatīts programmu izstrādes princips programmējot pielikumu *C++* programmēšanas valodā.

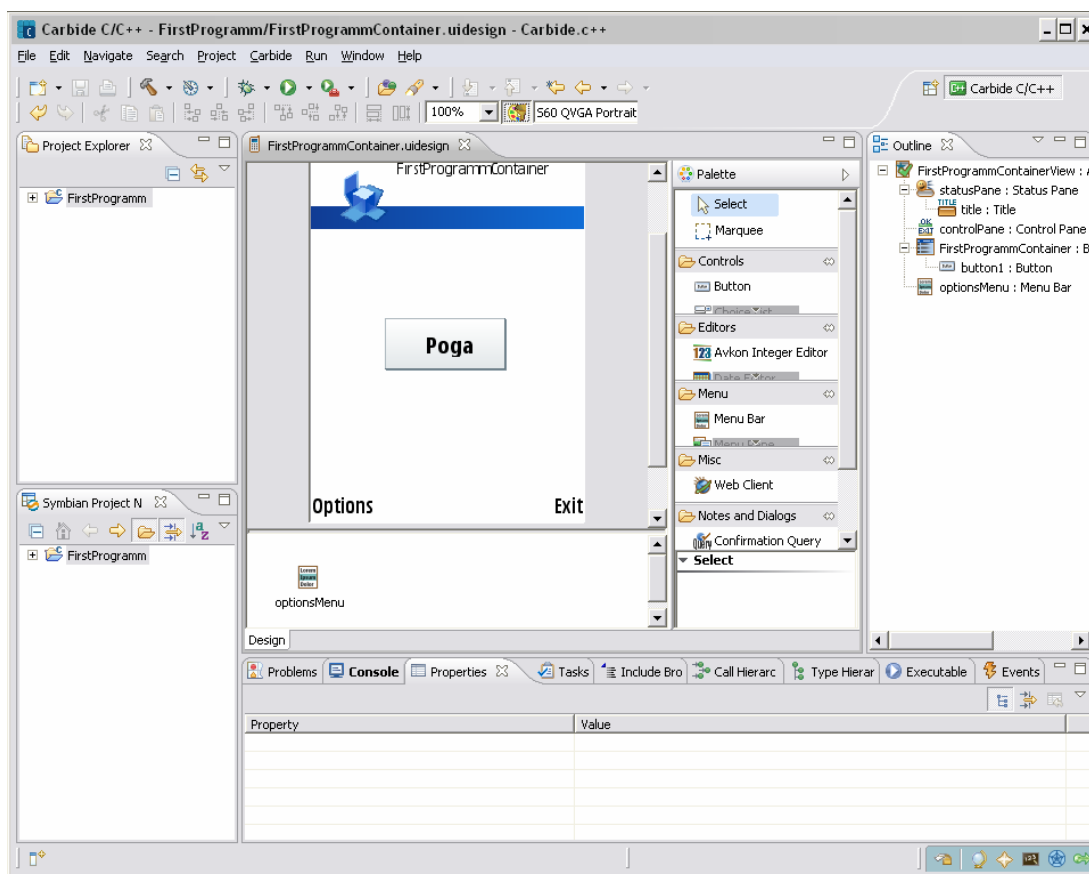
Lai varētu izstrādāt lietojumus priekš *Symbian OS v9.4 C++* programmēšanas valodā ir nepieciešamas sekojošas programmas (6):

- **Carbide C++** - ir Eclipse-bāzēta programmēšanas vide, kas atbalsta *Symbian OS* izstrādāšanu priekš *S60*, *Series 80*, *UIQ* un *MOAP* operētājsistēmām. Programma ir pieejama internetā kompānijas Nokia mājas lapā. Tā ir pieejama bez maksas.
- **GCC for Symbian** – ir kompilators, kas ļauj nokompilēt *C++* programmas kodu priekš *Symbian OS*.
- **Symbian Certificate Manager** – šī programma ir programmas *Carbide C++* spraudnis, un tā ļauj veidot izstrādātāja sertifikātus programmā *Carbide C++*.
- **Symbian DevCert Request** – šī programma ļauj iegūt *Symbian* izstrādātāja sertifikātu.
- **Symbian MakeSIS** – ir rīks, kāš ļauj programmas izstrādātājam sapakot savu programmas .sis failā, tā lai pēc tam šo failu var uzinstalēt mobilajā ierīcē ar *Symbian OS* operētājsistēmu.
- **Symbian UnSIS** – ir programma, kas ļauj atpakaļ *Symbian OS* instalācijas failu .sis speciālajā direktorijā.
- **SDK** – programmatūras izstrādāšanas rīki.

*Symbian OS* oficiālajā mājas lapā internetā ir pieejama plaša dokumentācija programmētājiem, kas vēlas sākt programmēt priekš mobilajām ierīcēm ar *Symbian OS* operētājsistēmu. Dokumentācija ir pieejama gan reālā laikā izmantojot interneta pieslēgumu, gan arī to ir iespējams lejupielādēt. *Symbian OS* operētājsistēmas mājas lapā dokumentācija ir pieejama tikai angļu valodā.

Lai varētu izveidot savu triviālu programmu priekš *Symbian OS v9.4* operētājsistēmas izmantojot *C++* programmēšanas valodu, nepieciešams uzstādīt uz datora visu nepieciešamo programmatūru (nepieciešamā programmatūra ir aprakstīta augstāk). Pēc tam nepieciešams:

- *Izveidot jaunu projektu.* Atveram programmēšanas vidi *Carbide C++*. Tālāk nepieciešams nospriest izvēlnē **File > New > Symbian OS C++ Project**. Pēc tam, logā, kas atveras, izvēlamies projekta tipu. Izvēlēsimies **GUI Application with UI Designer** un spiežam **Next**. Pēc tam ievadam projekta nosaukumu (piemēram: *FirstProgramm*). Tālāk var uzstādīt kompilēšanas parametrus (var neko nemainīt un atstāt kā ir). Pēc tam var izvēlēties programmas īpašības (var neko nemainīt un atstāt kā ir). Nākamajā solī izvēlamies sagatavi jaunajam projektam: paņemam sagatavi **Empty** un spiežam **Next**. Nākamajā solī spiežam pogu **Finish**.
- *Izveidojam nelielu funkcionālu.* Paletes joslā (**Palette**) atrodam kontroli **Button** un ar peli pārnesam to uz lietotāja interfeisa dizaina loga. Pēc tam ar peles kursoru pogas izmērus var izmantīt, pavelkot pogu aiz viena no tā stūriem. Lai izmainītu pogas tekstu, divas reizes jānospiež uz pogas, un tad ir jāieraksta teksts, kuru vajag ielikt pogā. Rezultāts ir redzams attēlā 2.3.



2.3. att. Programmas Carbide C++ programmas logs

- *Nokompilēt un pārbaudīt darbību.* Spiežam izvēlni **Run > Run** vai taustiņus uz klaviatūras **Ctrl+F11**. Ja programma veiksmīgi nokompilējās, tad pēc brīža atvērsies

mobilās ierīces emulatora logs un tajā ielādēsies nokompilētā programma. Attēlā 2.4. ir redzams triviālas programmas rezultāts mobilās iekārtas emulatorā.



#### **2.4. att. Triviālas programmas izpildes rezultāts Symbian OS v9.4 sistēmā**

Programmas veidošana notiek samērā vienkārši un ātri. Lai pārbaudītu, vai izveidotā programma darbojas korekti, nav nepieciešam mobilā ierīce. To var aizstāt ar mobilās iekārtas emulatoru, kas ir ļoti ērti tāpēc, ka pateicoties mobilās iekārtas emulatoram, programmatūras veidotājam var arī nebūt pati mobilā iekārt. Slikti ir tas, ka mobilās iekārtas emulatori ielādējas ļoti lēni (var paiet līdz pat 1 minūte, līdz emulatori pilnībā ielādējas un tajā tiek ielādēta jaunizveidotā programma).

## 2.3. Programmatūras uzstādīšana mobilajā iekārtā

Lai programmu uzinstalēt mobilajā iekārtā ar *Symbian OS*, sākumā to ir jāielādē mobilajā iekārtā. Veido, kā to izdarīt, ir daudz:

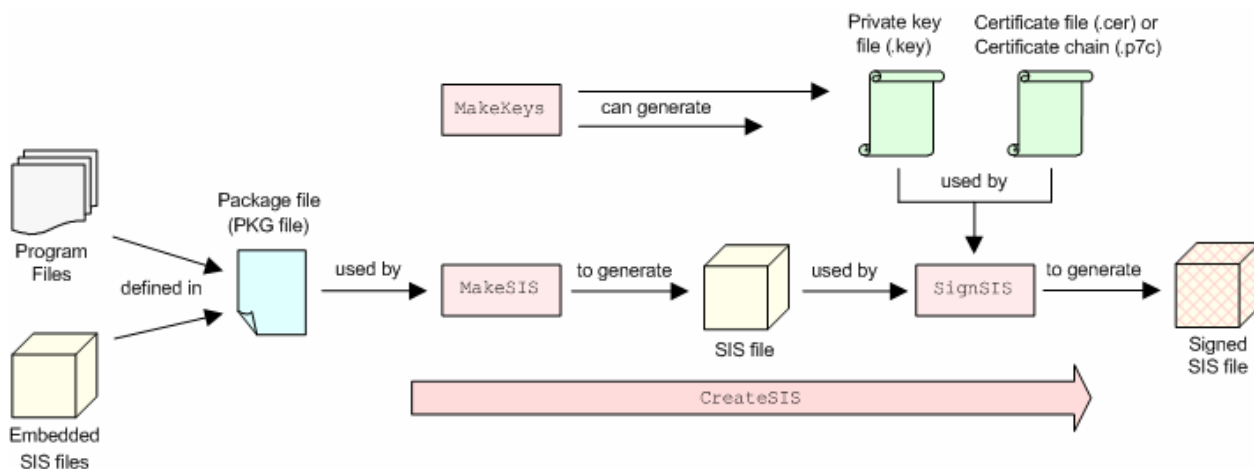
- Pieslēdzot mobilo iekārtu datoram ar speciālās programmatūras un aparatūru (ar vadu, *Bluetooth* vai *IR*);
- Nosūtot aplikāciju uz mobilo iekārtu no citas mobilās iekārtas ar īsziņas, *Bluetooth* vai *IR* palīdzību;
- Nosūtot aplikāciju uz mobilo iekārtu ar e-pasta palīdzību;
- Lejupielādējot aplikāciju mobilajā iekārtā no interneta;
- Ja mobilā iekārta atbalsta atmiņas kartes, tad izmantojot atmiņas karti, pārnest no datora aplikāciju uz mobilo iekārtu.

*Symbian OS* operētājsistēmā programmas var instalēt tikai no *.sis* (**sis** ir *Symbian OS Installation System*) failiem, tāpēc, lai izveidoto programmu varētu uzinstalēt mobilajā iekārtā ar *Symbian OS*, programmai ir nepieciešams izveidot *.sis* failu.

*CreateSIS* ir rīks, kas tie uzinstalēts datorā kopā ar *SDK* priekš programmu izstrādes mobilajām iekārtām ar *Symbian OS*, kas tiek lietots veidojot *.sis* instalācijas failus. *CreateSIS* ir vairāku programmu apkopojums:

- **MakeSIS** – izveido *.sis* failu no teksta-bāzēta *.pkg* faila;
- **SignSIS** – instalācijas faila piesaistītājs;
- **MakeKeys** – drošības sertifikātu veidotājs.

Šīs programmas var lietot arī atsevišķi. Attēlā 2.5. ir parādīts algoritms, kā tiek veidots *.sis* fails.



2.5. att. *.sis* faila veidošanas algoritms

Tātad, lai izveidotu .sis failu, sākumā jāizveido .pkg fails. .pkg ir teksta fails un tā formāts ir aprakstīts 1. pielikumā. Tad kad .pkg fails ir izveidots, tad izsaucam programmu *CreateSIS* kā parametru tai nododot failu .pkg. Programmas izpildes rezultātā saņemam .sis failu, kuru jau ir iespējams nosūtīt uz mobilo iekārtu ar *Symbian OS v9.4* operētājsistēmu ar kādu no augstāk aprakstītajiem veidiem (6).

### **3. Programmatūras izstrādes iespējas priekš mobilajām iekārtām ar iPhone OS operētājsistēmu**

*iPhone OS* ir samērā jauna mobilo iekārtu operētājsistēma. Pirmo reizi to izsludināja *MacWorld Expo* konferencē 9. janvārī 2007. gadā kopē ar *iPhone* telefonu. Pateicoties tam, ka *iPhone* telefons kļuva ļoti populārs sākumā ASV un pēc tam arī visā pasaulē, *iPhone OS* kļuva par vienu no populārākajām operētājsistēmām pasaules tirgū. Operētājsistēma tika speciāli izstrādāta *Apple* korporācijas *iPhone* mobilajam telefonam, optimizējot *Mac OS X* operētājsistēmu tā, lai tā varētu strādāt uz mobilā telefona (10).

### 3.1. Operētājsistēmas apraksts

Operētājsistēma *iPhone OS* ir izveidota speciāli priekš *Apple* korporācijas *iPhone* telefoniem. 2007. gada janvārī kopā ar pirmo *iPhone* telefonu ASV tirgū parādījās pirmā šīs operētājsistēmas versija. Bet no sākuma šai operētājsistēmai nebija oficiāla nosaukuma. Pirmo reizi tas parādījās, kad 2008. gada 6. martā tika izlaista pirmā *beta* versija *iPhone* izstrādātāju rīku kopai (*iPhone SDK*) (11).

Operētājsistēma ir sadalīta četros abstraktajos slāņos:

- Operētājsistēmas kodola slānis;
- Operētājsistēmas servisu slānis;
- Vides slānis;
- *Cocoa* skārienu kontroles un ekrāna grafikas slānis.

Operētājsistēma uz mobilās iekārtas aizņem 240 megabaitu no iekārtas atmiņas.

2008. gada 11. jūlijā kopā ar jaunā *iPhone 3G* iznākšanu parādījās jaunā operētājsistēmas versija *iPhone OS 2.0* (11). Galvenās operētājsistēmas atšķirības bija tādas, ka jaunajā *iPhone* telefonā tika ieviests 3G tīklu atbalsts un GPS modulis, līdz ar to operētājsistēmai vajadzēja strādāt ar šīm jaunajām iekārtām. Pirmā *iPhone OS 2.0* versija tika izlaista tirgū ar daudzām kļūdām, kārtīgi neiztestējot to. Tāpēc daudzi mēdiji daudz rakstīja par to, ka jaunais *Apple* izstrādājums nav labs. Bet pēc dažiem mēnešiem pēc kārtas tika izlaistas uzlabotās operētājsistēmas versijas.

Šī gada jūnijā tiek plānots izlaist jaunu operētājsistēmas versiju *iPhone OS 3.0*. Jaunajā operētājsistēmas versijā būs ap 100 jaunu funkciju ieskaitot: teksta kopēšanu un ielīmēšanu, MMS atbalsts, Spotlight, Speak Notes, peers to peers, īsziņu pārsūtīšanu un dzēšanu, kā arī daudz kas cits (11).

Operētājsistēmas pārinstalācija notiek ļoti vienkārši. *iPhone* telefonu pieslēdz ar USB vadu pie datora. Datora ar pieslēgumu pie interneta ieslēdz programmu *iTunes*. Ieslēdzot programmu *iTunes*, tā automātiski salīdzina pieslēgtā *iPhone* telefona operētājsistēmas versiju ar *Apple* korporācijas serveros pieejamo operētājsistēmas versiju. Ja ir pieejama jaunāka operētājsistēmas versija, tad programma piedāvā lejupielādēt un uzstādīt jaunāku operētājsistēmas versiju. Pēc lejupielādēšanas, *iTunes* programma saglabā uz datora visus datus no mobilās iekārtas un pārkopē uz to jaunāku operētājsistēmas versiju. Pēc tam, kad instalācija ir pabeigta, tad telefonā tiek pārkopēti visi lietotāja dati, kas bija saglabāti telefonā pirms operētājsistēmas pārinstalēšanas un




telefons tiek nokonfigurēts tāpat, kā tas bija pirms operētājsistēmas pārinstalēšanas. Tas ir ļoti vienkārši, jo lietotājam nekas nav jāzina par to, kā vajag pārinstalēt telefona operētājsistēmu. To var izdarīt jebkurš cilvēks, kurš mācēs pieslēgt telefonu pie datora ar USB vadu un palaist uz datora *iTunes* programmu.

Darbā tiek apskatīts *iPhone 3G* telefons ar 2.2.1 operētājsistēmas versiju, jo uz darba izstrādes brīdi šī ir pati jaunākā operētājsistēmas versija. *iPhone* telefonam nav klaviatūras un gandrīz visas darbības sistēmā tiek darītas ar skārienjūtīga ekrāna palīdzību. Operētājsistēmas galvenā izvēlne ir redzama attēlā 3.1.



3.1. att. iPhone OS operētājsistēmas galvenā izvēlne

Ekrāna augšējā joslā ir vairāki indikatori:

- GSM pieslēguma indikators - 
- Mobilo pakalpojumu operatora nosaukums - **AT&T**;
- Bezvadu tīkla pieslēguma indikators - 
- Pulkstenis - **00:00**;
- Akumulatora enerģijas daudzuma indikators - 

Pēc noklusējuma operētājsistēmā ir uzstādītas sekojošas programmas:

- Īsziņu sūtīšanas un lasīšanas programma. Programma ir realizēta tā, ka īsziņas netiek skatītas atsevišķi, bet gan tiek attēlota visa sarakste ar konkrētu personu no kontaktu saraksta.

- Kalendārs un plānotājs.
- Fotogrāfiju pārlūks.
- Programma darbam ar fotokameru.
- YouTube – programma, kas ļauj skatīties video ruļļus no interneta lapas [www.youtube.com](http://www.youtube.com).
- Akciju cenu pārlūks.
- Maps – programma, kas ļauj izmantot *Google* kartes, kā arī attēlot kartēs tagadējo *iPhone* telefona atrašanās vietu, izmantojot GPS moduli.
- Laika prognožu programma, kura nolasa laika prognozes no [www.weather.com](http://www.weather.com) interneta lapas.
- Pulkstenis, modinātājs, hronometrs un taimeris.
- Kalkulators.
- Pierakstu bločiņš.
- Telefona uzstādījumu rediģēšanas programma.
- iTunes – programma, kas ļauj pirkt mūzikas un video saturu no *iTunes* servisa internetā.
- App Store – programma, kas ļauj lejupielādēt maksas un bezmaksas programmas priekš *iPhone* telefona no interneta.
- Zvanišanas funkciju izmantošanas programma un kontaktu grāmata.
- E-pasta pārlūks.
- Safari – interneta pārlūks.
- iPod – programma, kas ļauj atskaņot audio un video saturu.

### 3.2. Programmatūras izstrādes iespējas, izmantojot operētājsistēmas izstrādātāja sniegtos rīkus

Oficiālā izstrādātāju rīku kopā *iPhone SDK* tika publicēta 2008. gada 6. martā. No šī brīža gandrīz katru mēnesi, reizēm pat dažas reizes mēnesī, tika izlaista jauna izstrādes rīku kopas versija. 2009. gada 17. martā tika izlaista pirmā izstrādes rīku kopā priekš *iPhone OS 3.0*, bet bakalaura darba izstrādes brīdī pati operētājsistēma *iPhone OS 3.0* vēl nav izlaista, tāpēc darbā tiks apskatīta izstrādes rīku kopa priekš *iPhone OS 2.2.1* versijas.

Lai varētu izstrādāt programmas priekš iPhone telefona, ir nepieciešams lejupielādēt no Apple oficiālās mājas lapas internetā izstrādes rīku kopu priekš vajadzīgās operētājsistēmas versijas. Izstrādes rīku kopai ir pieejami divi veidi:

- Standarta versija – versijas cena 99 USD;
- Uzņēmumu versija – versijas cena 299 USD.

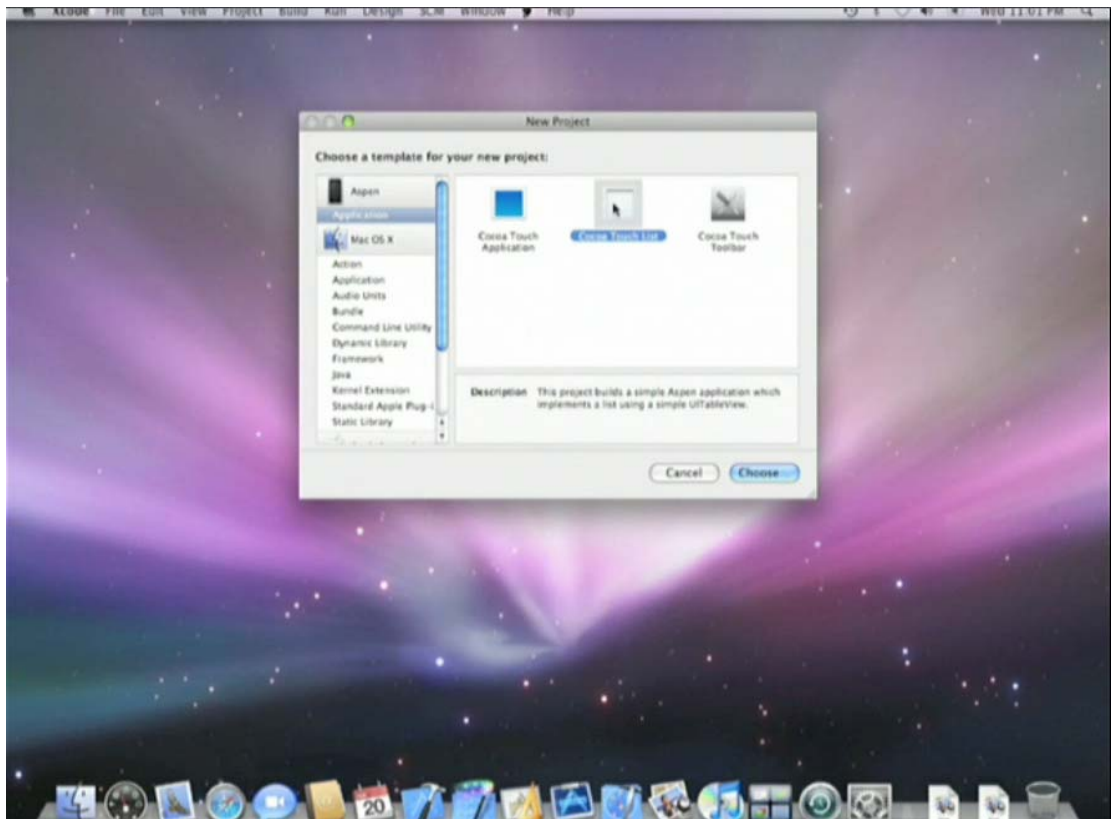
Izstrādātāju rīku kopa ir pieejama tikai priekš Mac OS X Leopard 10.5.4 vai jaunākas versijas (9).

*iPhone SDK* ietver sevī:

- XCode – programmēšanas vide, kas ļauj veidot projektus, rakstīt kodu *Objective-C* programmēšanas valodā un atklūdot programmas.
- Instruments – programma, kas ļauj mērīt dažādus sistēmas parametrus programmas izpildes laikā, kas palīdz optimizēt programmu ātrdarbību.
- Dashcode – programma, kas ļauj izveidot vizuālo programmu daļu, veidot programmu interfeisu, kā arī veidot Web bāzētas aplikācijas priekš *iPhone* telefona.
- Simulator – iPhone telefona emulators.

Lai izveidotu triviālu programmu priekš iPhone telefona, atveram XCode programmēšanas vidi:

- *Izveidojam jaunu projektu izmantojot sagatavi.* Augšējā izvēlnē spiežam **Project > New**. **New Project** logā kreisajā izvēlnē izvēlamies **Applications** un no piedāvātajām sagatavēm izvēlamies sagatavi **Cocoa Touch List**. Piemērs ir redzams attēlā 3.2.



### 3.2. att. Jauna projekta izveidošanas logs

Tad spiežam pogu **Choose**. Atveras logs, kurā ir jāieraksta jauna projekta nosaukums un jāizvēlas direktorijs, kurā saglabāt šo projektu.

- *Nokompilējam un izsaucam programmu*. Sagatave ir izveidota kā jau gatava programma, tāpēc programmēt kaut ko jaunu nav nepieciešams, tāpēc, lai paskatītos, kā stādā programma, spiežam pogu **Build and Go** no augšējās rīku joslas programmā *XCode*. Pēc veiksmīgas kompilēšanas programmēšanas vide *XCode* automātiski izsauks *iPhone* telefona emulatoru un palaidīs tajā tikko nokompilētu programmu. Piemērs ir redzams attēlā 3.3.



### 3.3. att. Jauna projekta izpilde iPhone telefona emulatorā

- *Apstādināt programmas izpildi.* XCode programmēšanas vidē rīku joslā nospiežam pogu **Tasks**.

Programmas piemērs ir paņemts no Apple korporācijas oficiālās mājas lapas internetā (9).

### 3.3. Programmatūras uzstādīšana mobilajā iekārtā

Ir divi veidi, kā programmatūru uzstādīt *iPhone* telefonā.

Pirmais, ir nepieciešams uzstādīt telefonā speciālu operētājsistēmu, kura ir domāta programmu testēšanai. To izdarīt ir vienkārši. Ierīce ir jāpieslēdz pie datora. Jāpalaiž programma *Organizer*. Galvenajā programmas logā izvēlnē **Summary** ir jāizvēlas operētājsistēmas versija, kuru vēlaties izmantot testēšanai un nospiegt pogu **Reset iPhone**. Programma pati uzstādīs visu. Tas ir ļoti ērti un vienkārši, jo tam nav vajadzīgas nekādas operētājsistēmas uzstādīšanas īpatnības, programma pati izpilda visu nepieciešamo. Pēc tam *XCode* programmēšanas vidē uzstādījumos jānomaina no *iPhone OS Simulator* uz *iPhone OS* un vienkārši palaižot programmu, tā tiks nokopēta mobilajā ierīcē, un to varēs lietot.

Otrs veids, kā uzlikt programmu *iPhone* telefonā ir lejupielādēt to no *App Store* servisa. Lai izveidoto programmu varētu augšupielādēt *App Store* servisā, lai citi lietotāji var to lejupielādēt bez maksas vai nopirkt, ir nepieciešama izstrādes rīku kopās uzņēmumu versija.

## 4. Salīdzināšana

Lai varētu izprast plusus un mīnus programmu izstrādāšanā augstāk aprakstītajām operētājsistēmām, ir nepieciešams veikt salīdzināšanu pēc šādiem kritērijiem:

- Oficiālo izstrādes rīku pieejamība;
- Oficiālo izstrādes rīku dokumentācijas pieejamība;
- Oficiālo izstrādes rīku uzstādīšanas sarežģītība;
- Programmu izstrādāšanas sarežģītība;
- Programmu testēšanas sarežģītība;
- Programmu uzstādīšanas sarežģītība mobilajā iekārtā.

Salīdzināšanas kritēriji tika izvēlēti tieši tādi tāpēc, lai varētu saprast cik grūti cilvēkam, kurš neko nezina par to, kā veidot programmatūru priekš mobilajām iekārtām, ir sākt tās taisīt.

Kritērijus bakalaura darba autors ir izdomājis pats.

*Windows Mobile* operētājsistēmas salīdzināšanas kritēriju apraksts ir redzams tabulā 4.1.

4.1. tabula

### Windows Mobile operētājsistēmas salīdzināšanas kritēriji

Oficiālo izstrādes rīku pieejamība	+ Oficiālā izstrādes rīku kopa ir pieejama lejupielādēšanai no interneta. - Ir pieejama bezmaksas versiju, bet bez maksas var lietot tikai 90 dienas. + Ir sadalījums pa dažādām izstrādātāju grupu kategorijām. - Izstrādes rīkus nav iespējams pasūtīt uz DVD diska.
Oficiālo izstrādes rīku dokumentācijas pieejamība	+ Dokumentācija ir pieejama internetā. + Dokumentācija ir ļoti plaša un ir domāta gan iesācējiem gan arī pieredzējušiem programmētājiem. - Dokumentācija ir pieejama tikai angļu valodā.
Oficiālo izstrādes rīku uzstādīšanas sarežģītība	+ Ir jāinstalē vairākas programmas, kuras ir nepieciešamas programmu veidošanai, bet oficiālajā mājas lapā ir skaidri aprakstīts, kādas programmas ir vajadzīgas. - Oficiālie izstrādes rīki ir pieejami lietošanai tikai <i>Windows</i>

	operētājsistēmās.
Programmu izstrādes sarežģītība	+ Programmēšanas vidē ir pieejamas visdažādākās sagataves, kas atvieglo programmētāja darbu.
Programmu testēšanas sarežģītība	+ Programmas var vienkārši testēt ar ierīces emulatoru. - Grūtāk programmas testēt ierīcē, jo katru reizi ir jāpārtaisa instalācijas faili un jākopē programma pa jauna mobilajā iekārtā.
Programmu uzstādīšanas sarežģītība mobilajā iekārtā	+ Ir jāizdara vairākas darbības, lai programmu varētu sagatavot kopēšanai mobilajā iekārtā, bet dokumentācijā ir skaidri aprakstītas visas darbības, kas ir jādara. - Programmas ielādēt operētājsistēmā var tikai ar speciālās programmatūras palīdzību.

*Symbian OS* salīdzināšanas kritēriju apraksts ir redzams tabulā 4.2.

4.2. tabula

#### Symbian OS salīdzināšanas kritēriji

Oficiālo izstrādes rīku pieejamība	+ Oficiālā izstrādes rīku kopa ir pieejama lejupielādēšanai no interneta. + Oficiālā izstrādes rīku kopa ir pieejama bez maksas. + Oficiālo izstrādes rīku kopa ļauj izstrādāt programmas izmantojot vairākas programmēšanas valodas. - Izstrādes rīkus nav iespējams pasūtīt uz DVD diska.
Oficiālo izstrādes rīku dokumentācijas pieejamība	+ Dokumentācija ir pieejama internetā. + Dokumentācija ir ļoti plaša un ir domāta gan iesācējiem gan arī pieredzējušiem programmētājiem. - Dokumentācija ir pieejama tikai angļu valodā.
Oficiālo izstrādes rīku uzstādīšanas sarežģītība	+ Ir jāinstalē vairākas programmas, kuras ir nepieciešamas programmu veidošanai, bet oficiālajā mājas lapā ir skaidri aprakstīts, kādas programmas ir vajadzīgas. - Oficiālie izstrādes rīki ir pieejami lietošanai tikai <i>Windows</i> operētājsistēmās.
Programmu izstrādes sarežģītība	+ Programmēšanas vidē ir pieejamas visdažādākās sagataves, kas atvieglo programmētāja darbu.

Programmu testēšanas sarežģītība	+ Programmas var vienkārši testēt ar ierīces emulatoru. - Grūtāk programmas testēt ierīcē, jo katru reizi ir jāpārtaisa instalācijas faili un jākopē programma pa jauna mobilajā iekārtā.
Programmu uzstādīšanas sarežģītība mobilajā iekārtā	+ Ir jāizdara vairākas darbības, lai programmu varētu sagatavot kopēšanai mobilajā iekārtā, bet dokumentācijā ir skaidri aprakstītas visas darbības, kas ir jādara. + Programmas ielādēt operētājsistēmā var dažādos veidos, gan izmantojot speciālo izstrādātāju programmatūru, gan arī bez tās.

*iPhone OS* salīdzināšanas kritēriju apraksts ir redzams tabulā 4.3.

4.3. tabula

#### **iPhone OS salīdzināšanas kritēriji**

Oficiālo izstrādes rīku pieejamība	+ Oficiālā izstrādes rīku kopa ir pieejama lejupielādēšanai no interneta. - Oficiālā izstrādes rīku kopa nav pieejama bez maksas. + Ir sadalījums pa dažādām izstrādātāju grupu kategorijām. - Izstrādes rīkus nav iespējams pasūtīt uz DVD diska.
Oficiālo izstrādes rīku dokumentācijas pieejamība	+ Dokumentācija ir pieejama internetā. + Dokumentācija ir ļoti plaša un ir domāta gan iesācējiem gan arī pieredzējušiem programmētājiem. - Dokumentācija ir pieejama tikai angļu valodā.
Oficiālo izstrādes rīku uzstādīšanas sarežģītība	+ Ir jāinstalē tikai viena programmu pakete, kura ietver sevī visas programmas, kura ir nepieciešama programmu veidošanai. - Oficiālie izstrādes rīki ir pieejami lietošanai tikai <i>Mac OS X</i> operētājsistēmās.
Programmu izstrādes sarežģītība	+ Programmēšanas vidē ir pieejamas visdažādākās sagataves, kas atvieglo programmētāja darbu.
Programmu testēšanas sarežģītība	+ Programmas var vienkārši testēt ar ierīces emulatoru. + Programmas var vienkārši testēt ar mobilo ierīci, jo pārkopēšana mobilajā ierīcē notiek nospiežot vienu pogu.
Programmu uzstādīšanas sarežģītība mobilajā	+ Programmas uzstādīšana notiek ar vienas pogas nospiešanu. + Programmas ielādēt operētājsistēmā var dažādos veidos, gan

iekārtā	izmantojot speciālo izstrādātāju programmatūru, gan arī bez tās.
---------	--

## Secinājumi

Izskatot visus salīdzināšanas kritērijus aprakstītajām mobilo iekārtu operētājsistēmām autors nonāca pie secinājuma, ka programmatūras izstrādāšanas iespējas ir apmēram vienādas. Dažos gadījumos kritēriju apraksti atšķiras dažādām operētājsistēmām, bet galvenokārt tie sakrīt. Kā vienu no galvenajiem mīnusiem visām operētājsistēmām var minēt to, ka izstrādātāja rīku kopas dokumentācijas ir pieejamas tikai angļu valodā un to, ka oficiālo izstrādes rīku kopās var lietot tikai vienā operētājsistēmā. Pozitīvi ir tas, ka oficiālās izstrādātāju rīku kopas ir pieejamas internetā un tas, ka tām ir pieejama laba dokumentācija. Kā arī pozitīvi ir tas, ka operētājsistēmu izstrādātāji ir parūpējušies par to, lai programmu izstrādātājiem būtu vienkāršāk izstrādāt lietojumus, un izveidoja daudz dažādas sagataves.

Autors pirmajā vietā liek programmu izstrādāšanu priekš *iPhone OS* operētājsistēmas, otrajā vietā programmu izstrādāšana *Windows Mobile* operētājsistēmai un trešajā programmu izstrādāšanu *Symbian OS* operētājsistēmai.

Pēc šī darba uzrakstītas autors secināja, ka kļūt par programmu izstrādātāju priekš kādas no aprakstītajām mobilo iekārtu operētājsistēmām ir samērā vienkārši.

Un tādēļ, ka mobilās iekārtas aizvien vairāk tiek pielietotas mūsdienās un tāpēc tām ir vajadzīgas aizvien jaunākas un jaunākas programmatūras, autoram liekas, ka būtu noderīgi izveidot kursu bakalaura studiju programmu studentiem, kurā tie varētu apgūt programmēšana prasmes priekš mobilajām iekārtām.

## Nobeigums

Darbā tika apskatītas trīs populārākās mobilo iekārtu operētājsistēmas: *Windows Mobile*, *Symbian OS* un *iPhone OS*, un tika apskatītas programmatūras izstrādāšanas iespējas izmantojot oficiālos izstrādes rīkus. Kā arī tika apskatīti veidi, kā izveidoto programmu var uzstādīt mobilajā iekārtā. Pēc tam tika salīdzinātas programmatūras izstrādāšanas iespējas aprakstītajās operētājsistēmas pēc vairākiem iepriekš aprunātiem salīdzināšanas kritērijiem. Beigās tika izteikti autora secinājumi pēc šī darba uzrakstīšanas.

Nākotnē šo darbu var turpināt, izpētot dziļāk operētājsistēmu izstrādātāju sniegtos rīkus pieejamās bibliotēkas un sagataves un salīdzināt programmatūras izstrādes iespējas jau koda līmenī, kā arī praktiski izstrādāt sarežģītākas programmas.

Izstrādājot šo darbu, autors ir sapratis, kā tiek veidotas programmatūras priekš dažādām mobilo iekārtu operētājsistēmām, uzzināja, kas tam ir vajadzīgs, kā arī praktiski ir pamēģinājis izstrādāt primitīvas programmas.

## Izmantotā literatūra un avoti

1. *Windows Mobile for Developers | Windows Mobile Overview* [tiešsaiste] Pieejams: <http://developer.windowsmobile.com/>
2. *MSDN: Microsoft Development, MSDN Subscriptions, Resources, and more* [tiešsaiste] Pieejams: <http://msdn.microsoft.com/>
3. *Microsoft Corporation* [tiešsaiste] Pieejams: <http://www.microsoft.com/>
4. *Windows Mobile – Википедия* [tiešsaiste] Pieejams: <http://ru.wikipedia.org/wiki/WindowsMobile/>
5. *Mobile-review.com обзор операционной системы Windows Mobile 6 Professional* [tiešsaiste] Pieejams: <http://www.mobile-review.com/pda/articles/wm-crossbow.shtml>
6. *Symbian Developer Network* [tiešsaiste] Pieejams: <http://developer.symbian.com/>
7. *Symbian OS – Википедия* [tiešsaiste] Pieejams: [http://ru.wikipedia.org/wiki/Symbian\\_OS/](http://ru.wikipedia.org/wiki/Symbian_OS/)
8. *Forum Nokia – Driving innovation for mobile application developers* [tiešsaiste] Pieejams: <http://www.forum.nokia.com/>
9. *iPhone Dev Center* [tiešsaiste] Pieejams: <http://developer.apple.com/iphone/>
10. *iPhone – Википедия* [tiešsaiste] Pieejams: <http://ru.wikipedia.org/wiki/IPhone/>
11. *iPhone OS – Wikipedia, the free encyclopedia* [tiešsaiste] Pieejams: [http://en.wikipedia.org/wiki/IPhone\\_OS/](http://en.wikipedia.org/wiki/IPhone_OS/)

## Pielikumi

### 1. pielikums. .pkg faila formāts

```
; FirstProgram.pkg
; Rindiņas, kas sākas ar semikolu, tiek uzskatītas par komentāru
rindiņu.

; Valodas, kuras tiek atbalstītas
; none - tikai Angļu pēc noklusējuma

; Saraksts ar lokalizētiem pārdevēju nosaukumiem. Vismaz viena ir
jābūt obligāti (Angļu [EN]).
; Sarakstam ir jāatbilst valodu sarakstam, kurš ir aprakstīts .pkg
failā
%{"Symbian"}

; Nelokalizēts pārdevēja nosaukums
:"Symbian"

; Paketes galvene
; Programmas nosaukums, Paketes UID (var izveidot šeit:
https://www.symbiansigned.com/), versija, numurs, kompilēšanas
numurs, paketes veids
#{"FirstProgram"}, (0xE80000A6), 1, 0, 0, TYPE=SA

; Produkts priekš S60 platformas
; Produkta/platformas versijas UID, versija(priekš S60 netiek
izmantots), numurs(priekš S60 netiek izmantots), kompilēšanas
numurs(priekš S60 netiek izmantots), Produkta ID
[0x101F7961], 0, 0, 0, {"S60ProductID"}

; Faili, kuri ir jāuzinstalē no manas direktorijas
; izpildāmie faili ir jākopē direktorijā \sys\bin
"c:\S60\devices\Nokia_N97_SDK_v0.5\epoc32\release\gcce\urel\First
Program.exe"-":\sys\bin\FirstProgram.exe"
; UI resursu faili ir jākopē direktorijā \Resource\Apps\
"c:\S60\devices\Nokia_N97_SDK_v0.5\epoc32\data\Z\Resource\Apps\Fi
rstprogram.rsc"-":\Resource\Apps\Firstprogram.rsc"
; reģistrācijas faili ir jākopē direktorijā
\private\10003a3f\import\apps\
"c:\S60\devices\Nokia_N97_SDK_v0.5\epoc32\data\z\Private\10003a3f
\Apps\Firstprogram_reg.rsc"-
":\private\10003a3f\import\apps\Firstprogram_reg.rsc"
```

## Bakalaura darbs

Mobilu iekārtu programmatūras izstrādes iespēju salīdzināšana dažādās operētājsistēmās

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai. Piekrītu sava darba publicēšanai internetā.

Autors: \_\_\_\_\_  
(Autora paraksts)

Ar savu parakstu apliecinu, ka esmu lasījis augšminēto bakalaura darbu un atzīstu to par **piemērotu/nepiemērotu** (nevajadzīgo svītrot) aizstāvēšanai Latvijas Universitātes datorzinātņu bakalaura studiju programmas gala pārbaudījuma komisijas sēdē.

Darba vadītājs(-ja): \_\_\_\_\_  
(Vadītāja paraksts)

Darbs iesniegts Datorikas fakultātē \_\_\_\_\_  
(Iesniegšanas datums)

Ar šo es apliecinu, ka darba elektroniskā versija ir augšupielādēta LU informatīvajā sistēmā.  
Metodiķe: \_\_\_\_\_  
(Metodiķes paraksts)

Recenzents: Leo Seļāvo, Dr. dat., asociētais profesors \_\_\_\_\_

Darbs aizstāvēts bakalaura darbu gala pārbaudījuma komisijas sēdē

\_\_\_\_\_ prot. Nr. \_\_\_\_\_, vērtējums \_\_\_\_\_  
(Darba aizstāvēšanas datums)

Komisijas sekretārs: \_\_\_\_\_  
(Sekretāra paraksts)