



Міністерство освіти і науки України
ЗНТУ

Кафедра програмних засобів

**"РОЗРОБКА ЗАСТОСУВАНЬ ДЛЯ
МОБІЛЬНИХ ПРИСТРОЇВ"**

Конспект лекцій

для студентів спеціальності

122

Комп'ютерні науки та інформаційні технології

усіх форм навчання

Запоріжжя 2016

ЗМІСТ

ВСТУП.....	3
ПРИНЦИП РОБОТИ МОБІЛЬНОГО АПАРАТУ	4
ПРИСТРІЙ МОБІЛЬНОГО ТЕЛЕФОНУ І ЙОГО ОСНОВНІ ФУНКЦІОНАЛЬНІ ВУЗЛИ (МОДУЛІ).	5
ПРИЗНАЧЕННЯ Й РОБОТА ОКРЕМИХ ВУЗЛІВ.	7
I. Комп'ютерні програми та їх мобільні аналоги.....	14
1.1 Основні характеристики	14
1.2 Ключові моменти при розробці мобільних додатків	17
1.3 Перспективи розвитку.....	23
II. Найпопулярніші мобільні операційні системи та їх характеристика	25
2.1 Google Android System	25
2.2 Apple IOS	30
2.3 Windows Phone	35
III. Розробка мобільного мультимедіа додатку на базі платформи Android..	42
IV. Розробка додатків для мобільних платформ iphone, android, blackberry ..	50
4.1. Вступ	50
4.2. Розробка мобільних додатків.....	51
4.3. Мобільні додатки.....	52
4.4. Мови й інструментальні засоби.....	53
V. Інструментарій кросплатформної розробки.....	55
VI. Мобільні пристрої в курсах комп'ютерних наук	56
VII. Мобільні платформи як об'єкт наукових досліджень	58
VIII. Висновки	60
Література.....	61

ВСТУП

Без стільникового зв'язку сучасне людство не може уявити собі й дня. І буквально зовсім недавно головними функціями були тільки дзвінки та відправка повідомлень. Однак сьогодні пріоритети дещо змінюються. Телефон став далеко не просто засобом для комунікації людей, але і багатофункціональним пристроєм, що дозволяє пізнавати світ, проводити дозвілля і навіть заробляти гроші. І, звичайно, все це стало можливо тільки завдяки розвитку Інтернету для мобільних пристроїв і створених з цієї нагоди додатків. У XXI столітті почався бурхливий розвиток ринку мобільного контенту, зокрема, мобільних додатків. Здешевити мобільний інтернет трафік дозволило поява нових технологій - GPRS і EDGE. Користувачі починають викачувати із Всесвітньої мережі величезна кількість картинок, ігор, рингтонів і пр.

Ринок мобільних пристроїв стали заповнювати смартфони і комунікатори. Вони відрізнялися від звичайних мобільних телефонів наявністю досить розвиненою операційною системою (Windows Phone, Symbian OS, RIM, Android, Mac OS), а також більш широкими можливостями і більшою продуктивністю.

З'являються спеціалізовані сайти з продажу контенту для мобільних пристроїв.

Зараз, крім різних маркетів додатків, подібні веб-ресурси створюють провідні виробники стільникових телефонів, смартфонів і комунікаторів.

В даний час спеціалізовані портали з розповсюдження апп мають відомі по всьому світу корпорації: гігант Apple (App Store), компанія Google (Google Play), компанія Nokia (OVi), виробник смартфонів Blackberry - компанія RIM (BlackBerry App World і Application Center), компанія Sony Ericsson (PlayNow arena) і ряд інших.

Крім самих додатків, ці онлайн-ресурси продають також різноманітний мобільний контент (музику, відео, картинки, електронні книги і т.д.).

ПРИНЦИП РОБОТИ МОБІЛЬНОГО ТЕЛЕФОНУ

У теоретичній частині я не буду глибшатися в історію створення стільникового зв'язку, про її засновників, хронологію стандартів і т.д. Кому це цікаво – матеріалу предосить як у друкованих виданнях, так і в мережі інтернет.

Розглянемо, що ж із себе представляє мобільний (стільниковий) телефон.

Стільниковий телефон – це передавач^-передавач-приймально-передавач, що працює на одній із частот у діапазоні 850МГц, 900МГц, 1800МГц, 1900МГц. Причому приймання й передача рознесені по частотах.

Система GSM складається з 3-х основних компонентів, таких як:

- підсистема базових станцій (BSS – Base Station Subsystem);
- підсистема перемикання/комутації (NSS – Networkswitchingsubsystem);
- центр керування й обслуговування (OMC – Operation and Maintenance Centre);

Двома словами працює це так:

Стільниковий (мобільний) телефон взаємодіє з мережею базових станцій (БС). Вишки БС звичайно встановлюють або на своїх наземних щоглах, або на дахах будинків або інших споруджень, або ж на орендованих уже існуючих вишках усіляких ретрансляторів радіо/ТВ і т.п., а також на висотних трубах котелен і інших промислових споруджень.

Телефон після включення й усе інше час моніторить (прослуховує, сканує) ефір на наявність GsmнСигналу своєї базової станції. Сигнал своєї мережі телефон визначає по спеціальному ідентифікатору. Якщо такий є (телефон перебуває в зоні покриття мережі), то телефон вибирає кращу за рівнем сигналу частоту й на цій частоті посилає БС запит на реєстрацію в мережі.

Процес реєстрації по суті є процесом аутентифікації (авторизації). Його суть полягає в тому, що кожна SimaКарта, вставлена в телефон, має свої

унікальні ідентифікатори IMSI (International Mobile Subscriber Identity) і Ki (Key for Identification). Ці самі IMSI і Ki заносяться в базу центру аутентифікації (Auc) при випуску виготовлених SimkКарт операторів зв'язки. При реєстрації телефону в мережі ідентифікатори передаються БС, а саме Auc. Далі Auc (центр ідентифікації) передає телефону деяке випадкове число, яке є ключем для виконання обчислень по спеціальному алгоритму. Це обчислення відбувається одночасно в мобільному телефоні й Auc, після чого обое результату рівняються. Якщо вони збігаються, то SimaКарта з'являється справжньою й телефон реєструється в мережі.

Для телефону ж ідентифікатором у мережі є його унікальний номер IMEI (International Mobile Equipment Identity). Цей номер звичайно складається з 15 цифр у десятковій виставі. Наприклад 35366300/758647/0. Перші вісім цифр описують модель телефону і його походження. Що залишилися – серійний номер телефону й контрольне число.

Даний номер зберігається в енергонезалежній пам'яті телефону. У застарілих моделях цей номер можна перемінити за допомогою спеціального програмного забезпечення (ПО) і відповідного програматора (іноді й датакабелю), а в сучасних телефонах він дублюється. Один екземпляр номера зберігається в області пам'яті, яку можна програмувати, а дублікат – у зоні пам'яті OTP (One Time Programming), яка програмується виробником один раз і не має можливості перепрограмування.

Отож, якщо навіть змінити номер у першій області пам'яті, то телефон, при включенні, порівнює дані обох областей пам'яті, і, якщо виявляються різні номери IMEI – телефон блокується. Для чого все це міняти, запитаете ви? Насправді законодавство більшості країн забороняє це робити. Телефон по номеру IMEI відслідковується в мережі. Відповідно при крадіжці телефону його можна відстежити й вилучити. А якщо встигнути змінити цей номер на будь-який інший (робітник), те шанси знайти телефон зводяться до нуля. Цими питаннями займаються спецслужби при відповідній допомозі оператора мережі і т.д. Тому глибокшати в цю тему не стану. Нас цікавить чисто технічний момент зміни номера IMEI.

Справа в тому, що при певних обставинах даний номер може ушкодитися в результаті збою ПО або неправильного його відновлення й тоді телефон абсолютно не придатний для експлуатації. От отут на допомогу й приходять усі засоби, щоб відновити IMEI і працездатність апарата.

Докладніше цей момент буде розглянутий у розділі програмного ремонту телефону.

Тепер коротенько про передачу голосу від абонента до абонента в стандарті GSM. Насправді це технічно дуже складний процес, який абсолютно відрізняється від звичної передачі голосу по аналогових мережах як, наприклад, домашній провідний/радіо телефон. Чимсь віддалено схожі цифрові DecteРадіотелефони, але реалізація однаково інша.

Справа в тому, що голос абонента, перш ніж буде переданий в ефір, зазнає безлічі перетворень. Аналоговий сигнал розбивається на відрізки тривалістю 20мс, після чого перетворюється в цифровий, після чого кодується шляхом застосування алгоритмів шифрування з т.зв. відкритим ключем – система EFR (Enhanced Full Rate - удосконалена система кодування мови, розроблена фінською компанією Nokia).

Усі сигнали кодека обробляються дуже корисним алгоритмом на основі принципу DTX (Discontinuous Transmission) – переривчастої передачі мови. Його корисність полягає в тому, що він управляє передавачем телефону, включаючи його тільки в тому момент, коли починається вимова мови й відключає в паузах між розмовою. Усе це досягається за допомогою включеного в кодек VAD (Voice Activated Detector) –детектор активності мови.

У прийнятого абонента всі перетворення відбуваються у зворотному порядку.

Пристрій мобільного телефону і його основні функціональні вузли (модулі).

Будь-який мобільний телефон – це складний технічний пристрій, що полягає з безлічі функціонально закінчених модулів, які взаємозалежні між собою й у цілому забезпечують нормальну роботу апарата. Вихід з ладу хоча б одного модуля спричиняє мінімум – часткову несправність апарата, максимум – телефон повністю непридатний.

Призначення й робота окремих вузлів.

1. Акумуляторна батарея (АКБ) – основний (первинний) джерело живлення телефону. У процесі експлуатації має одна неприємна властивість – старіння, тобто втрата ємності, збільшення внутрішнього опору. Це необоротний

процес і швидкість старіння акумулятора залежить від багатьох факторів, ключовими з яких є правильна експлуатація й зберігання.

Раніше основна маса АКБ для телефонів проводилася за технологіями NiCd (на основі нікелю й кадмію), NiMH (нікель-металгідрид). У цей час дані акумулятори зняті з виробництва. З поширенням АКБ на основі технології Li-Ion (літійіон), останні показали краще співвідношення ціна/якість, а також мали ряд переваг, зокрема відсутність т.зв. «ефекту пам'яті». Тривалість терміну служби становить приблизно 3-4 року. Не дуже давно на ринку з'явилися Li-Pol (літіймполімерні) акумулятори. Вони коштують дешевше літійіонних, але термін служби в них теж менше – приблизно 2 року.

Сучасні АКБ зізнаються працездатними, якщо в них збереглося не менш 80% від номінальної ємності. На практиці ж зустрічаються АКБ із 50% і менше. Тобто багато користувачів намагаються «вичавити» з акумулятора останні міліампери, через що самі потім і страждають, тому що нерідко зношений акумулятор починає здуватися, що може приводити до поломок корпусу телефону, а іноді навіть до виходу з ладу мережного зарядного пристрою, ланцюгів зарядки телефону, контролера живлення. Так що, на АКБ грошей заощаджувати не варто. Телефону теж потрібно гарне живлення.

Особливого відходу АКБ не вимагають. Головне, не допускати переохолодження в зимовий час (до -10°C), тому що прискорюється розряд і старіння. А так само нагрівання до $50-60^{\circ}\text{C}$ і вище. Це небезпечно – АКБ може попросту здутися й навіть вибухнути (саме для літєвих АКБ це критично)!!!

АКБ мобільного телефону складається з 2-х частин: властиво батареї й маленької плати електроніки-автоматики.

Що стосується плати електроніки, то вона виконує захисну функцію, запобігаючи як саму батарею, так і телефон від позаштатних ситуацій, таких як:

- коротке замикання (КЗ) живильних клем акумулятора;
- перегрів батареї в процесі зарядки й експлуатації;
- розряд батареї нижче встановленої мінімально припустимої норми;
- перезаряд батареї;

При виникненні однієї з них, спрацьовує т.зв. електронне реле й вихідні клеми АКБ знеструмлюються.

Як правило, сучасна АКБ має мінімум 3 контактних виводу для підключення до батарейного рознімання мобільного телефону. Це відповідно «+», «-», і «TEMP» (датчик температури, за допомогою якого контролер батареї разом з контролером живлення телефону управляють процесом

зарядки батареї, зменшуючи або збільшуючи зарядний струм, а при перегріві або КЗ взагалі відключають батарею від клем плати електроніки).

Слід зазначити, що в різних виробників розташування контактів може відрізнятися!!!

Основними характеристиками АКБ є:

- номінальна напруга – як правило 3,6 – 3,7Вольт. Для повністю зарядженого акумулятора 4,2 – 4,3 Вольт.
- ємність – для сучасних телефонів приблизно від 700мА до 2000мА й більш.
- внутрішній опір - чим менше - тем краще (приблизно до 200 миллиом)

2. Контролер живлення – служить для перетворення напруги АКБ у кілька видів напруг для живлення окремих вузлів і пристроїв телефону, таких, як CPU (центральний процесор), RAM і ROM (мікросхеми пам'яті), усіляких підсилювачів, іноді підсвічувань клавіатури й дисплея і т.д., а так само управляє процесом зарядки АКБ. Разом із процесором активує вбудовані в нього або ж зовнішні підсилювачі звуку розмовного динаміка, мікрофона, буззера (поліфонічного гучномовця). Плюс до всього забезпечує обмін даними з SimpКартою.

Конструктивно виконаний у вигляді окремого чипа. Іноді може бути сполучений із процесором (китайські підробки відомих брендів типу Nokia N95 і т.д.)

При нормальній експлуатації телефону контролер живлення рідко виходить із ладу. Найчастіше це трапляється під час зарядки при перегріві або при використанні неоригінального або несправного зарядного пристрою(ЗУ). Рідше - якщо телефон піддався впливу вологи, був сильно ударений.

Зовнішній вигляд представлений на мал.2 і може відрізнятися (залежить від конкретної моделі телефону і його виробника).

3. Sim-holder (sim – коннектор) – тримач SIM – карти. Виходячи з назви – служить для підключення SIM – карти до телефону. Конструкція практично однакова для всіх телефонів, тому що сучасні SIM – карти наведені до одного стандарту. Має в собі 6 (рідко 8) подпружинених контактів, за допомогою яких здійснюється електричний зв'язок SIM – карти й контролера живлення або процесора. Відрізняються лише конструкцією кріплення (утримання) SIM – карти. До поломок можна віднести облом контактів при частій зміні SIM – карт або ж недотепному (неправильному) їх добуванні, коли користувач починає застосовувати підручні засоби для підковирювання SIM – карти для подальшого захвату пальцями й добування із тримача. Часто до цього прибігають наші прекрасні дами, використовуючи свої довгі, з дорогим манікюром нігті. У підсумку – страждає й телефон і манікюр

Спеціального відходу коннектор не вимагає. Але бувають випадки (знову таки залежить від користувача), коли контакти окисняються, засмічуються, втрачають свої пружні властивості. У такому випадку допускається ДУЖЕ ОБЕРЕЖНО!!! протерти їх гумкою (ластиком) і ДУЖЕ ОБЕРЕЖНО!!!, злегка, голкою або дерев'яною зубочисткою підігнути контакти нагору.

При описаних вище несправностях SIM – холдера (тримача), телефон не буде «бачити» вашу SIM – карту й постійно буде виводити на дисплей повідомлення типу: «Вставте SIM – карту». Зламані тримачі ремонту не підлягають і вимагають заміни на нові.

4. Мікрофон – служить для перетворення голосу користувача в слабкі електричні сигнали з метою їх подальшого посилення, перетворення й відправлення в ефір. У стільникових телефонах бувають двох типів: аналогові й цифрові. Останні мають більш складну конструкцію й вимагають більше працездат при демонтажі й заміні.

Мікрофони втрачають свої експлуатаційні характеристики або виходять із ладу в основному при забрудненні, влученні води, при ударах телефону (особливо це стосується цифрових мікрофонів, тому що вони самі по собі дуже тендітні).

При несправностях мікрофона в телефоні можуть бути такі дефекти:

- другий абонент не чує користувача взагалі;
- другий абонент чує користувача дуже слабо;
- у слуховому (розмовному) динаміці чутний тріск (т.зв. наведення GSM – сигналу). Такий же шум можна почути, коли підносимо стільниковий телефон у режимі розмови або відправлення sms до працюючого радіоприймача, підсилювача, комп'ютерних колонок і т.д. Як прищепило, мікрофони не ремонтуються й підлягають заміні (крім випадків засмічення отворів, звуководів корпусу мобільного телефону. Їх слід просто очистити від пилу, бруду і т.д.)

5. Динамік (розмовний динамік)– служить для перетворення електричних сигналів у звукові коливання. Тобто працює у зворотному порядку мікрофона. Один абонент говорить у мікрофон, який перетворює голос в ел. сигнали, далі ці сигнали перетворюються (див. опис вище), випромінюються в ефір. Другий абонент ухвалює ці сигнали телефоном і чує їх у динаміку телефону.

У більшості телефонів встановлено кілька динаміків – окремо розмовний і окремо поліфонічний. Поліфонічний динамік відтворює мелодію при вхідному виклику, смс і т.д. Але є телефони (у більшості фірми Samsung), де роль розмовного й поліфонічного виконує той самий динамік. Тільки при відтворенні мелодії або інших сигналів активується додатковий підсилювач

потужності звуку. До несправностей динаміків можна віднести часткову несправність і повну. Часткова – це відтворення мови або музики дуже тихо, із хрипами й неприємним дзенькотом. Це можна усунути, але лише в тих випадках, коли, після зовнішнього огляду буде видно, що динамік засмічений сторонніми предметами. Наприклад такими, як дуже дрібна металева стружка, яка любить проникати через спеціально відведені отвори для виходу звуку динаміка. Це обумовлене тим, що динамік у своїй конструкції містить постійний магніт. От він і примагнічує до себе дрібні металеві предмети. Особисто я прихильник заміни таких динаміків на нові. По-перше, це заощадить вам час, який ви будете витратити на чищення, а його вам знадобиться чимало. По-друге, рідко буває, що після чищення динамік працює так само чисто, без викривлень і так само голосно. Так що, не думайте – відразу міняйте на новий. Особливо, якщо це телефон не ваш, а прийшов у ремонт.

Повна – відсутність звуку взагалі. Причина – обрив проведення звукової котушки динаміка. Вирішується тільки заміною динаміка. Про те, як перевірити динамік на справність (цілісність) я напишу нижче.

6. Спікер(буззер, дзвінок, поліфонічний динамік – це все те саме) – той же динамік, тільки в більшості випадків призначений для відтворення мелодії дзвінка, смс, MP3 і т.д. Але, як говорилося вище, може використовуватися й для розмови. Несправності й способи усунення такі ж, як і для розмовного динаміка.

7. Центральний процесор (CPU) – є основним пристроєм мобільного телефону. Це той же процесор, який є присутнім у будь-якому персональному комп'ютері, ноутбукі і т.д., тільки трошки поменше й примітивніше. Призначений для виконання машинних команд, інструкцій і операцій, передбачених програмним забезпеченням (прошиванням –розм.) телефону, а також чіткої взаємодії з іншими модулями й пристроями й наступного керування ними. Одним словом, процесор – це «мозок», який повністю управляє роботою мобільного телефону. Конструктивно виконаний у вигляді окремого чипа. Відповідає за безліч процесів, що відбуваються під час нормальної роботи телефону. Основні з них це: вивід зображення на дисплей, приймання й обробка сигналів стільникової мережі, приймання й обробка сигналів клавіатурного модуля, керування роботою камери, пристроїв приймання/передачі інформації, процесом зарядки акумулятора (разом з контролером живлення) і багато іншого.

За умови нормальної експлуатації телефону процесор практично ніколи не виходить із ладу й ніякого відходу не вимагає.

У сучасних телефонах, а особливо смартфонах (у перекладі з англ. смартфон – розумний телефон. Той же телефон, тільки має подібність із комп'ютером у виді наявності операційної системи й безліччю встановлюваних програм для виконання тих або інших завдань) часто встановлюється 2 процесора. Один з них виконує ті ж функції, що й у звичайному телефоні, а другий призначений для роботи операційної системи й виконання її програм.

При виході з ладу центрального процесора телефон повністю непрацездатний.

8. Flash – пам'ять. Окремий чип (мікросхема), який призначений для зберігання програмного забезпечення телефону (прошивання, *firmware*), а так само даних користувача (контакти, мелодії, фотографії і т.д.). Програмне забезпечення (прошивання, *firmware*) – це розроблена виробником телефону програма, яка обробляється й виконується процесором. Для користувача – це те, що він бачить на екрані мобільного телефону й ті функції, які йому доступні в конкретній моделі телефону.

Флеш Пам'ять так само рідко виходить із ладу за умови нормальної експлуатації. Але слід пам'ятати, що ці чипи мають хоч і велике, але все-таки обмежена кількість циклів читання/запису інформації.

Флеш Пам'ять є енергонезалежною й зберігає всі записані в неї дані навіть після відключення джерела живлення (наприклад, АКБ).

9. RAM – пам'ять (ОЗУ). Служить для тимчасового зберігання даних. У ній проводяться всі процесорні обчислення програмного коду, а також зберігаються результати обчислень і обробки інформації в конкретний теперішній момент (наприклад, прослуховування музики, відтворення відео, робота додатків, ігор і т.д.) Через непотрібність пам'ять очищається від одних даних і завантажує нові й так постійно.

Слід пам'ятати, що пам'ять ОЗУ (оперативний запам'ятовувальний пристрій) є енергозалежною і у випадку відключення джерела живлення всі дані, які зберігалися в ОЗУ будуть загублені!!!

10. Клавіатурний модуль – стандартна цифрова клавіатура для набору номера абонента, тексту смс повідомлень + набір додаткових кнопок, які виконують певні програмним забезпеченням телефону функції, наприклад регулювання рівня гучності, запуск програм, фотокамери, диктофона і т.д. Для нормальної роботи клавіатурного модуля основне завдання користувача – містити клавіатуру в чистоті й не допускати влучення вологи, бруди й інших предметів. А якщо ні, то кнопки доводиться давити з більшим зусиллям або ж телефон взагалі не реагує на натискання. Відновити роботу клавіатурного модуля можна методом чищення від забруднень. Якщо ж контактні майданчики й з'єднуючі їхні провідники були піддані впливу

вологи або ін. рідин і були ушкоджені, то такий клав.модуль підлягає заміні на новий.

11. LCD –дисплей – властиво дисплей (екран) телефону. Призначення всім зрозуміле, тому глибшатися на цьому не стану. Основними характеристиками є такі параметри, як:

- розв'язна здатність, тобто кількість відтворених пікселів (крапок). Чим вище цей параметр, тем чіткіше і якісніше буде картинка. Для більш-менш сучасних телефонів властиві такі дозволи екрана: 220X176 пікселів, 320X240. Для телефонів з більшими сенсорними екранами: 400X240, 640X360, 800X400.

- кількість відтворених (відображуваних) кольорів. Теж саме, чим більше, тем краще. У застарілих телефонах з кольоровими дисплеями це значення в основному 4096 кольорів. У міру вдосконалювання цей параметр збільшився до 65тис., потім досягся 262тис.. Зараз усі сучасні дорогі телефони постачені дисплеями із глибиною кольору 16млн.

При правильній експлуатації телефону дисплей не вимагає ніякого відходу. У деяких випадках, коли телефон використовується в запиленім середовищі або ж просто згодом у корпус набилося багато пилу й сміття, то дисплей необхідно АКУРАТНО протерти мікрофіброю (спеціальна протиральна серветка, яка добре очищає й не залишає слідів і розлучень. Її можна придбати в салонах продажу оптики. Деякі види окулярів комплектуються такою протиральною мікрофіброю.) При експлуатації телефону не можна допускати фізичного впливу на дисплей (удари, здавлювання, сильні перегини), а також піддавати впливу прямих сонячних променів і підвищеної температури. Це приведе до виходу його з ладу.

12. Приємопередатчик – служить для приймання й передачі стільникового GsmнСигналу. Містить у собі багато функціональних елементів (генератори керовані напругою приймача й передавача, смугові фільтри, що розв'язують конденсатори, індуктивності і т.д.). Управляється процесором і кварцовим резонатором 26Мгц.

При несправностяхприємопередатчика телефон не зможе зареєструватися в стільниковій мережі й на дисплеї буде відсутній індикатор рівня GsmнСигналу.

13. Підсилювач потужності – призначений для посилення сигналу, вироблюваногоприємопередатчиком, до рівня потужності, необхідного для випромінювання антеною в ефір.

При несправностях підсилювача потужності телефон буде ухвалювати сигнал стільникової мережі, але зареєструватися в ній не зможе, тому що не зможе передавати GsmГСигнал.

14. Антенний перемикач (світч) – призначений для сполучення (підключення) прийомного й передавального тракту GsmМодуля до антени телефону. Цим досягається наявність у телефоні однієї загальної антени для приймання й передачі, а також виключається вплив підсилювача потужності на прийомний тракт.

Конструктивно антенні перемикачі виконані на тендітній керамічній підложці й, при падіннях телефону, дуже часто виходять із ладу, тому що підложка тріскається. У таких випадках телефон «не бачить» сигнал стільникової мережі.

15. Антена – призначена для нагромадження енергії, випромінюваною базовою станцією й наступної передачі її в ланцюзі прийомного тракту. При передачі сигналу всі навпаки: з передавача сигнал підсилюється підсилювачем потужності й подається в антену, яка випромінює сигнал в ефір.

I. КОМП'ЮТЕРНІ ПРОГРАМИ ТА ЇХ МОБІЛЬНІ АНАЛОГИ

1.1 Основні характеристики

Найбільш істотні відмінності між додатками для мобільних пристроїв і додатками для настільних комп'ютерів обумовлені, ймовірно, тим, яким саме чином люди їх використовують. Найчисленніша категорія користувачів використовує комп'ютер в основному для наступних видів роботи: перегляд ресурсів в Web, робота з документами (обробка текстових документів, електронних таблиць, малюнків і тому подібного) та спілкування (електронна пошта, миттєвий обмін повідомленнями і так далі). У типових випадках всі ці види діяльності носять довготривалий, дослідницький характер, і користувач періодично переходить від одного з цих основних занять до іншого.

У більшості випадків способи використання настільного і мобільного варіантів одного і того ж додатка відрізняються між собою, хоча і доповнюють один одного. Як правило, робота користувача з мобільним додатком носить характер короткочасних сеансів взаємодії, яке або переривається ззовні, або саме викликає переривання сеансу. Зазвичай користувач або реагує на спробу перервати його роботу, або використовує пристрій для негайної передачі запиту певній особі або процесу. Чудовою ілюстрацією цього може служити типова практика використання сучасних мобільних телефонів. Коли ви здійснюєте телефонний дзвінок або відправляєте текстове SMS-повідомлення, ваш абонент отримує сигнал переривання; якщо ж дзвонять або надсилають SMS -повідомленням вам, то переривається ваша робота. Ця ж модель використання переноситься і на будь-яке інший додаток, що виконується на мобільних пристроях. Мобільний додаток може вважатися вдало розробленим лише в тому випадку, якщо користуватися ним настільки ж просто і природно, як зв'язуватися з кимось по телефону або обмінюватися SMS-повідомленнями.

Крім того, додаткам для мобільних пристроїв властива яскраво виражена спеціалізація, що забезпечує максимально зручний доступ користувача до обмеженого набору певних можливостей, що й відрізняє пристрої від тієї універсальної дослідницької середовища, створення якої характерно для успішних додатків, що виконуються на настільних обчислювальних машинах. Оскільки управління мобільними пристроями найчастіше здійснюється однією рукою або за допомогою невеликого екрану, дуже важливо забезпечити користувача можливостями оперативної ідентифікації та швидкого знаходження потрібної йому інформації чи коштів. Можливість швидкої навігації в межах нечисленного набору ключових засобів є важливим показником якості розробки мобільного додатку.

Робота за настільними комп'ютерами, як правило, ведеться протягом тривалих робочих сеансів. Самими звичайними є годинні сеанси, однак нерідко вони затягуються на кілька годин. У процесі тривалої роботи формуються і розробляються ідеї, а сама робота носить ітеративний характер з властивими йому поверненнями до вже пройдених етапів. Тому більше значення має не скорочення тривалості підготовки до роботи, а надання користувачеві багатого набору різноманітних засобів, які можуть йому знадобитися в процесі дослідження та обробки відповідної інформації.

Лептопи в основному використовуються так само, як і настільні комп'ютери, однак і в цьому випадку вже можна помітити ознаки, характерні для використання мобільних пристроїв. Більше значення надається скороченню часу запуску (або пробудження) програми та можливості швидкого переходу користувача до тієї роботи, якою він займався до завершення попереднього робочого сеансу. Разом з тим, за характером їх використання лептопи ближче до настільних комп'ютерів, ніж до мобільних пристроїв, що частково пояснюється просто близькістю фізичних розмірів відповідних категорій устаткування. Поки ви дістанете комп'ютер із сумки, відкриєте кришку, запуститься комп'ютер, доберетесь до додатка з яким збираєтесь працювати, або підключитесь, якщо це необхідно, до мережі

інтернет, користувач мобільного телефону встигне вирішити масу завдань. Тому ноутбуки дуже далекі від моделі типу "дістань пристрій з кишені і працюй".

Що ж стосується мобільних пристроїв, то аналогічні дії користувач може виконати набагато швидше, і в типових випадках на це йде від декількох секунд до декількох хвилин. Такі операції, як виконання телефонного дзвінка, перегляд розкладу зустрічей або відправка і читання повідомлень, здійснюються в процесі часто повторюваних, але короткочасних сеансів. У разі мобільних пристроїв навіть ігри відносяться до короткочасних видів діяльності. Зазвичай мобільні пристрої використовуються для ігор протягом коротких проміжків часу в перервах між виконанням інших операцій. Ігри призначені для того, щоб допомогти людям скоротати час в очікуванні поїзда, довгого перебування в аеропорту або в очікуванні партнера який спізнюється на зустріч з вами. Саме фактори короткочасності використання і можливості негайного доступу до необхідних засобів і змушують нас тримати мобільні пристрої в "постійно включеному" стані або в стані "негайної готовності до включення", в яких час, необхідний для отримання доступу до пристрою, можна порівняти з часом, який йде на вилучення пристрою з кишені.

Важливо відзначити, що незважаючи на короткочасність робочих сеансів, дані, з якими працює користувач, нерідко мають довготривалий характер. Користувач мобільного пристрою може захотіти продовжити гру з того місця, на якому вона була перервана під час попереднього сеансу.

У процесі ведення переговорів по телефону або під час особистих зустрічей часто доводиться звертатися до розкладу зустрічей і заносити в нього дату і час чергової зустрічі. Користувач повинен мати можливість швидко отримати доступ до пристрою, перейти до списку особистих контактів, створити новий пункт розкладу та одразу ж продовжити ведення переговорів без будь-якої затримки. Відповідні дані залишаються актуальними протягом тривалого часу і повинні зберігатися довго, але сам

додаток використовується протягом коротких періодів часу в режимі негайного доступу до інформації.

1.2 Ключові моменти при розробці мобільних додатків

Порівнянні і протиставлення мобільних пристроїв і їх додатків з їх настільними і серверними аналогами ми відвели в цьому розділі досить багато місця і часу. Тому зараз буде вельми доречно перерахувати все те, що робить мобільний додаток дійсно якісним.

- Час запуску.

Важливою характеристикою мобільних додатків є час їх запуску. Оскільки мобільними пристроями зазвичай користуються часто і протягом нетривалих проміжків часу, здатність мобільного додатку до швидкого запуску є обов'язковою вимогою. Можливо, 6-секундне споглядання заставки текстового процесора, енциклопедії або середовища розробки в очікуванні появи основного вікна відповідної програми і приносить мало задоволення, однак, з урахуванням загальної тривалості типового сеансу роботи з такими додатками, цей фактор можна вважати малозначним.

У разі ж мобільного застосування, яке користувач збирається використовувати протягом 20 секунд для уточнення або поновлення невеликої порції інформації, втрата 6 секунд представляється недозволеною розкішшю. Розумно керуватися мнемонічним правилом , згідно з яким тривалість сеансу роботи користувача з додатком повинна набагато перевершувати тривалість запуску цього додатка. У разі додатків для настільних комп'ютерів робочі сеанси тривають довше, і тому користувачі готові миритися з періодами очікування більшої тривалості. У разі ж мобільних додатків тривалість робочих сеансів менше, і тому висувається вимога , щоб тривалість періоду запуску програми була відповідно меншою. Тривалість періоду запуску додатка є важливим чинником і у випадку настільних комп'ютерів, але у випадку мобільних пристроїв, які

характеризуються нерегулярністю і короткочасністю робочих сеансів, цей чинник набуває вирішальне значення.

- Відгук пристрої.

Сприймаючи мобільні пристрої як слухняні механічні іграшки, люди очікують від них відповідної поведінки. Коли користувач впливає на маніпулятор, натискає на кнопку або будь-яким іншим чином маніпулює елементами керування пристроєм, він розраховує на фізичну реакцію з його боку. Не отримавши негайної реакції з боку пристрою, нетерплячий користувач починає нервувати і тут же повторює спробу виконання потрібної операції.

Якщо повторна спроба маніпуляції елементом управління, виконання клацання чи іншого аналогічного дії будуть оброблені вашим додатком або, помилково, іншим додатком, то в результаті цього можуть виникнути проблеми. Тому вкрай важливо дбати про те, щоб після виконання яких-небудь дій з пристроєм користувач отримував від нього негайне підтвердження реакції на ці дії в тій чи іншій формі.

Ідеальним варіантом такого підтвердження є виконання запитаної операції; краще цього нічого бути не може. На другому місці в цьому відношенні стоїть підтвердження того, що запит отримано і обробляється у фоновому режимі, в той час як додаток зберігає здатність сприймати інші запити. Третє місце належить відображенню курсора очікування, який подаватиметься користувача про те, що запит обробляється; додаток ні на які подальші запити не реагує, але користувач знає, що робота з обслуговування його запиту розпочато і виконується. Найгірший з варіантів - це коли користувач не бачить ніякої відповідної реакції з боку пристрою і йому залишається лише здогадуватися про те, сприйнятий його запит чи ні. Як і у випадку інших характеристик, які ми будемо вивчати, ці вимоги не є унікальними для мобільних пристроїв, але в даному випадку вони набувають особливого значення в силу специфіки роботи з цими пристроями і того, чого люди від них чекають.

- Фокусування уваги на окремих завданнях.

Ще однією характеристикою мобільного додатку, що визначає його успішність, є ступінь його спеціалізації. Для програми повинен бути чітко визначений набір завдань, ефективне вирішення яких воно має забезпечувати; здійснюються при цьому операції повинні виконуватися швидко і вимагати лише мінімальної кількості клацань, натискань або інших дій з боку користувача.

Саме з цієї причини в пристроях часто передбачаються спеціальні кнопки, що відповідають окремим завданням, які здатне вирішувати даний пристрій (наприклад, кнопка для переходу безпосередньо до бази даних, що зберігає інформацію про контакти, кнопка для перегляду календаря зустрічей або кнопка для зчитування штрих- коду та передачі його певного додатка).

Ви повинні докласти максимум зусиль до того, щоб якомога точніше визначити коло тих часто зустрічаються завдань, вирішення яких має вимагати мінімуму дій з боку користувача. Цією рекомендацією слід керуватися як щодо високорівневих функцій програми, так і щодо низькорівневих завдань, які зазвичай доводиться вирішувати користувачам в процесі роботи з даним додатком. Наприклад, якщо мобільний додаток має забезпечувати швидке введення дат, то ви повинні подбати про максимальне спрощення (і прискоренні) зазначеного процесу і вимірюванні його ефективності. Якщо ваш мобільний додаток повинен забезпечувати можливість вказівки певного місця розташування за допомогою схеми міських вулиць, щоб користувач міг вибрати маршрут руху до кінцевого пункту призначення, то ви повинні подбати про те, щоб це могло бути зроблено якомога швидше.

Одна з поширених помилок, що допускаються розробниками при створенні додатків для мобільних пристроїв, полягає в тому, що програмний код намагаються зробити якомога більш коротким, щоб тим самим максимально зменшити розмір програми. Хоча подібні цілі і є благородними, вони не повинні досягатися за рахунок зниження продуктивності праці

користувачів. Не шкодуйте часу на написання додаткового коду, фокусируючого увагу користувачів на виконанні спеціалізованих завдань і що дозволяє скоротити час отримання кінцевого результату

- Налаштування взаємодії із зовнішніми джерелами інформації.

Дуже важливо розуміти, що створення відмінного мобільного додатку - це не лише написання коду, який правильно виконується на пристрої. Не слід забувати також про зовнішній програмному забезпеченні, з яким взаємодіє ваше мобільний додаток. Необхідно, щоб у додатку відповідним чином враховувалися особливості інформаційних джерел, що обслуговують мобільні пристрої, і здійснювалася перевірка того, що інформація надається в тій формі, яка відповідає даному пристрою. Як приклад можна привести служби електронної пошти для мобільних пристроїв. Багатофункціональні поштові додатки вимагають наявності відповідного програмного забезпечення як на стороні сервера, так і на стороні клієнта. Клієнт здійснює доступ до сервера для отримання інформації щодо повідомлень, що надійшли і завантажує відповідний вміст в локальний пристрій. Оскільки мережеві з'єднання, використовувані мобільними пристроями, відрізняються нерегулярністю доступу до них, зниженою пропускну здатністю, а в багатьох випадках і більш високою вартістю в порівнянні з тими сполуками, які використовуються персональними комп'ютерами, поштові служби сервера повинні бути пристосовані для задоволення цих вимог. Зазвичай це досягається за рахунок того, що на сервері надаються кошти, що дозволяють обмежувати обсяг отриманого вмісту, або фільтри, пропускають лише ту інформацію, яка дійсно може знадобитися користувачеві мобільного пристрою. Для забезпечення підтримки ефективних сценаріїв при обслуговуванні мобільних пристроїв може знадобитися розширення функціональних можливостей тих служб на стороні сервера, які спочатку призначалися для обслуговування персональних комп'ютерів. Крім того, повинні бути передбачені такі механізми змін параметрів конфігурації програми для його виконання на серверах, настільних комп'ютерах і самих

мобільних пристроях, за допомогою яких користувачі зможуть визначати свої інформаційні запити і налаштовувати інформаційні фільтри таким чином, щоб забезпечувалося виконання відповідних вимог. Проектування мобільних додатків далеко не обмежується фізичними межами самих мобільних пристроїв.

- Однаковості стилю інтерфейсу.

З урахуванням того, що будь-який мобільний пристрій відрізняється компактністю і самодостатністю, бажання користувачів працювати з інтерфейсом, який не залежить від характеру розв'язуваних за допомогою пристрою завдань, видаються цілком природними. Кожне мобільний пристрій має власний функціональний колорит. Типове вдало спроектоване додаток сприймається не як розрізнений набір засобів, а як гармонійне розширення можливостей самого пристрою.

З цієї причини при проектуванні додатків для мобільних пристроїв дуже важливо дотримуватися єдиного стилю. Способи запуску і зупинки додатків, переміщення в межах інтерфейсу, а також організації діалогів між користувачем і додатками повинні ретельно продумувати окремо для кожного типу пристроїв. Користувачі мобільних пристроїв підсвідомо підлаштовуються під метафори користувача інтерфейсу, і будь-які відхилення від звичних шаблонів доставляють їм відчутні незручності. Створення звичної робочого середовища багато значить і в разі додатків для настільних комп'ютерів, але в силу різноманітності можливостей, які пропонуються такими додатками, вони дозволяють добитися однієї і тієї ж мети декількома способами (наприклад, за допомогою сполучень клавіш, клацань мишею, меню і панелей інструментів). У разі ж мобільних пристроїв для вирішення даної задачі часто є тільки один шлях, і користувач мимоволі звикає до одного певного способу. Набагато краще мати чотири різних версії додатка, кожна з яких відповідає певній метафорі користувача інтерфейсу конкретного пристрою, ніж одне загальне додаток, який не може бути інтегроване належним чином ні в одне цільове пристрій.

- Відмінності в архітектурі комп'ютерів.

Якщо скористатися аналогією з області архітектури, то настільні та переносні комп'ютери можна уподібнити великим садибам у сільській місцевості. Тоді відносно мобільних пристроїв можна сказати, що вони схожі на окремі чи комунальні міські квартири. Кожен з цих типів жител призначений для задоволення певних потреб і висуває специфічні вимоги щодо найбільш ефективних способів їх використання.

Зазвичай садиби характеризуються великими розмірами, і в них є багато місць, придатних для зберігання різних речей. Одні з таких місць знаходяться неподалік, тоді як дістатися до інших дещо важче. У садибах рідко використовувані речі зазвичай зберігаються на горищах або в підвалах, звідки їх можна періодично витягувати по мірі того, як в них виникає необхідність.

З іншого боку, в умовах міських квартир місць, придатних для зберігання речей, значно менше. У квартирах зберігають речі, які повинні бути завжди під рукою. Ті ж речі, якими користуються рідко, в квартирі не тримають. Замість того щоб купувати рідко використовувані речі, їх часто вигідніше орендувати.

Те ж саме справедливо і щодо мобільних пристроїв. Як оперативна пам'ять, так і пам'ять для довготривалого зберігання даних орієнтовані на зберігання тієї інформації, яка використовується найчастіше. Рідко використовувані дані повинні виштовхувати на сервери, звідки їх можна буде запросити, коли вони будуть потрібні.

У настільних комп'ютерах ОЗУ і довготривала пам'ять розділені. Водночас, в пристроях ОЗУ часто використовуються і як робоча пам'ять додатків, і як проміжна або довгострокова пам'ять. В якості довгострокових сховищ даних все частіше використовується флеш-пам'ять, зазвичай у вигляді знімних карт пам'яті.

1.3 Перспективи розвитку

Можливості мобільних додатків давно вийшли за рамки індустрії розваг, і їх використання дозволяє придбати додаткові переваги у сфері бізнес інтересів. Мобільні інформаційні технології сприяють поліпшенню сервісного обслуговування споживчої аудиторії, а також розвитку торгових і виробничих відносин. Крім того, не варто скидати з рахунків комерційний потенціал розважальних додатків, цей ринок як і раніше надзвичайно привабливий і має зростаючі перспективи. Доцільність замовлення мобільних додатків для бізнесу можна обґрунтувати відразу декількома факторами:

Користувачі інтернету (потенційні споживачі) все частіше виходять в мережу за допомогою мобільних пристроїв, а ряд додатків дозволяє фактично прив'язати до себе клієнтів і цільову аудиторію.

Можливість підключатися до серверів, які працюють за галузевим стандартам, і бути на зв'язку з корпоративними мережами забезпечують практично повну незалежність та оперативне реагування при вирішенні нагальних питань.

Динаміка розвитку ринку мобільних додатків підтверджує прогнози про високий зростання доходів в короткостроковій перспективі. Причому це не черговий банківський «мильна бульбашка», роздутий з чергових махінацій, а реальний продукт, який здатний принести відмінну прибуток.

Забезпечується більш тісний контакт з представниками найбільш платоспроможною цільовою аудиторії, так як в основному саме вони є власниками багатфункціональних мобільних пристроїв.

Якщо подивитися на сучасні напівпровідникові технології, то ми практично дійшли до меж можливого. Зараз шар діелектрика в транзисторі становить 10-15 атомів. Теоретична межа - це п'ять-шість атомів, і практично у розвитку традиційної напівпровідникової електроніки ми вже до нього наблизилися. Тому майбутнє за новими технологіями, і такі нові технології

розвиваються в надрах лабораторій IBM. Одне з інноваційних напрямів - це наноелектроніка. У 1998 році вчені Дональд Ейтлер і Ерхард Швейцер з лабораторії IBM в Каліфорнії зуміли викласти 35 атомами ксенону на кристалі нікелю логотип IBM. Це одне з перших звершень у галузі нанотехнології, коли людство навчилося керувати структурами фактично на атомарному рівні.

Подальший розвиток цей напрям отримав з винаходом так званих кремнієвих трубок. Кремнієва трубка на атомарному рівні може бути наповнена якимось вмістом і змінює свої якості залежно від геометрії і того, яким матеріалом вона наповнена. Розмір такої трубки становить 1-1,5 нанометра. Фактично це дозволяє створювати інтегральні мікросхеми на принципово іншому рівні і в принципово іншій щільності.

Причому ці нанотехнології вже вийшли за межі теоретичних розробок. Так, в лабораторії IBM в Швейцарії вчений Герд Біннінг, який отримав Нобелівську премію за розробку електронного мікроскопа, звернув увагу на можливість формування в певних полімерах маленьких ямочок, які мають розмір нанометрів. І мало того, ці ямочки можна як створювати, так і сканувати, що дозволило створити принципово нову технологію запису інформації. Щільність запису на таких пристроях пам'яті становить порядку 1 Тбайт на квадратний дюйм, це 25 мільйонів друкованих аркушів на кристалі розміром з поштову марку. Тобто на одному маленькому чіпі пам'яті, створеному за цією технологією, можна записати 15-20 Гбайт інформації. І цього року таке запам'ятовуючий пристрій було показано на виставці. Це не означає, що воно вже сьогодні готове до промислового випуску, але я думаю, що через два- три роки він вже буде можливий.

Існують і інші основи для розвитку інформаційних технологій. Такі, наприклад, як квантова обробка інформації, роботи з якої теж ведуться в надрах лабораторій IBM.

II. НАЙПОПУЛЯРНІШІ МОБІЛЬНІ ОПЕРАЦІЙНІ СИСТЕМИ ТА ЇХ ХАРАКТЕРИСТИКА

2.1 Google Android System

Android - операційна система для смартфонів, планшетів і нетбуків. Компанія Google придбала розробника програмного забезпечення Android inc. в 2005 році. Операційна система Android заснована на модифікованому ядрі Linux. Згодом, Google та інші учасники Open Headset Alliance співпрацювали для спільної розробки цієї нової операційної системи. Далі Android Open Source Project (AOSP) доручено підтримання та подальший розвиток платформи. У Android є велика спільнота розробників, які розширюють функціональність пристроїв.

OS Android має свій офіційний магазин з продажу додатків - Android Market. Включає він в себе як платні програми, так і безкоштовні. В даний момент, для України доступні для завантаження тільки безкоштовні програми та ігри. Так як OS Android є відкритою, користувачеві надається можливість завантажувати додатки та з інших ресурсів.

Під Android розробники, в основному, пишуть програми на мові Java, що керують пристроєм через розроблені Google бібліотеки.

Офіційно про OS Android стало відомо 5 листопада 2007 року, коли було оголошено підставу Open Headset Alliance - консорціум з 80 компаній. Велику частину коду Android була випущена під ліцензією Apache.

Android програми включають в себе java -додатки та бібліотеки, які запускаються віртуальною машиною Dalvik з JIT компілятором. Бібліотеки включають в себе систему управління, графіку OpenGL ES 2.0, движок WebKit, графічний движок SGL, SSL і бібліотеки Bionic. OS Android складається з 12 мільйонів рядків коду, в тому числі 3 -х мільйонів рядків XML, 2.8 мільйонів рядків на C, 2.1 мільйона рядків на Java і 1.75 мільйона рядків на C + +. Компанія Android inc. була заснована в жовтні 2003 року в

Пало Альто, штат Каліфорнія. Засновниками Android inc. були Енді Рубін, Річ Майнер, Нік Сірс і Кріс Уайт.

У серпні 2005 року компанія Google придбала Android inc. Після цього Android inc. стала дочірньою компанією Google. Після покупки Енді Рубін, Річ Майнер і Кріс Уайт залишилися в Android inc. Після поглинання Android inc., В мережі почали з'являтися чутки про те, що Google хоче вийти на ринок мобільних телефонів.

Отримавши підтримку Google, команда на чолі з Енді Рубіном почала працювати над операційною системою базується на ядрі Linux. Тоді ж, у грудні 2006 року поповзли чутки про те, що Google планує випустити смартфон під свої брендом, так званий " Гуглофон ".

Всі ці чутки спростував Ерік Шмідт, заявивши наступне: « Сьогоднішня заява носить більш амбітний характер, ніж випуск смартфона під нашим брендом, про що преса спекулює останні тижні. Наше бачення полягає в тому, що найпотужніша платформа повинна віддати свої сили тисячам різних моделей телефонів ». Після цих слів 5 листопада 2007 і був представлений Open Handset Alliance - консорціум з безлічі компаній, до якого увійшли такі гіганти як: Broadcom Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, T-Mobile, Texas Instruments та інші.

Назви кожної чергової версії ОС Android представляє собою назву якого-небудь десерту. Перші букви найменувань в порядку версій відповідають літерами латинського алфавіту.

Android Market - інтернет -магазин додатків для смартфонів на базі Android, яку просуває альянс Open Handset Alliance (ОНА) на чолі з Google. Включає в себе ігри, клієнти соціальних мереж, офісні додатки, додатки для читання новин, управління фінансами та інші.

22 жовтня 2008 Google оголосила про відкриття цього онлайн-магазину додатків для Android.

- В Україні на смартфонах перший час був відсутній офіційний магазин додатків. Ситуація була виправлена 12 січня 2010, коли про це повідомили в Samsung Україна, природно, що пізніше Android Market з'явився і у інших вендорів - Motorola, HTC, LG і Sony Ericsson, що значно підігріло інтерес до Android - смартфонам в Україні.

Першим офіційним смартфоном з Android Market в Україні став Samsung i5700 Galaxy Spica.

Програми для Android є програмами в нестандартному байт-кодi для віртуальної машини Dalvik.

Google пропонує для вільного скачування інструментарій для розробки (Android SDK), який призначений для x86-машин під операційними системами Windows XP, Windows Vista, Mac OS X (10.4.8 або вище) і Linux. Для розробки потрібно JDK 5 або JDK 6.

Розробку додатків для Android можна вести на мові Java (не нижче Java 1.5) . Існує плагін для Eclipse - « Android Development Tools » (ADT), призначений для Eclipse версій 3.3-3.5. Для IntelliJ IDEA також існує плагін, який полегшує розробку Android - додатків. Повідомляється, що для середовища розробки NetBeans IDE розроблено експериментальний плагін.



Рис. 2. 1. Архитектура OS Android

- Рівень додатків (Applications):

До складу Android входить комплект базових додатків: клієнти електронної пошти та SMS, календар, різні карти , браузер, програма для управління контактами і багато іншого. Всі додатки, що запускаються на платформі Android написані на мові Java.

- Рівень каркаса додатків (Application Framework):

Android дозволяє використовувати всю міць API, використовуваного в додатках ядра. Архітектура побудована таким чином, що будь-який додаток може використовувати вже реалізовані можливості іншої програми за умови, що останнє відкриє доступ на використання своєї функціональності. Таким чином, архітектура реалізує принцип багаторазового використання компонентів ОС і додатків.

Основою всіх додатків є набір систем і служб.

1. Система уявлень (View System) - це багатий набір уявлень з розширюваною функціональністю, який служить для побудови зовнішнього вигляду додатків, що включає такі компоненти, як списки, таблиці, поля введення, кнопки і т.п.

2. Контент - провайдери (Content Providers) - це служби, які дозволяють додаткам отримувати доступ до даних інших додатків, а також надавати доступ до своїх даних.

3. Менеджер ресурсів (Resource Manager) призначений для доступу до строковим, графічним та іншим типам ресурсів.

4. Менеджер сповіщень (Notification Manager) дозволяє будь-якому додатком відображати для користувача повідомлення в рядку статусу.

5. Менеджер дій (Activity Manager) управляє життєвим циклом додатків і надає систему навігації по історії роботи з діями.

- Рівень бібліотек (Libraries):

Платформа Android включає набір C / C + + бібліотек, використовуваних різними компонентами ОС. Для розробників доступ до

функцій цих бібліотек реалізований через використання Application Framework. Нижче представлені деякі з них:

1. System C library - BSD - реалізація стандартної системної бібліотеки C (libc) для вбудованих пристроїв, заснованих на Linux.
2. Media Libraries - бібліотеки, засновані на PacketVideo's OpenCORE , призначені для підтримки програвання і запису популярних аудіо -та відео-форматів (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG і т.п.).
3. Surface Manager - менеджер поверхонь управляє доступом до підсистеми відображення 2D - і 3D- графічних шарів.
4. LibWebCore - сучасний движок web -браузера, який надає всю міць вбудованого Android- браузера.
5. SGL - движок для роботи з 2D -графікою.
6. 3D libraries - движок для роботи з 3D - графікою, заснований на OpenGL ES 1.0 API.
7. FreeType - бібліотека, призначена для роботи зі шрифтами.
8. SQLite - потужний легковаговий движок для роботи з реляційними БД.

- Рівень середовища виконання (Android Runtime):

До складу Android входить набір бібліотек ядра, які надають велику частину функціональності бібліотек ядра мови Java.

Платформа використовує оптимізовану, реєстр - орієнтовану віртуальну машину Dalvik, на відміну від неї стандартна віртуальна машина Java - стек-орієнтована. Кожна програма запускається у своєму власному процесі, зі своїм власним примірником віртуальної машини. Dalvik використовує формат Dalvik Executable (*. Dex), оптимізований для мінімального використання пам'яті додатком. Це забезпечується такими базовими функціями ядра Linux, як організація потокової обробки і низькорівневе управління пам'яттю. Байт -код Java, на якому написані ваші програми, компілюються в dex - формат за допомогою утиліти dx, що входить до складу SDK.

- Рівень ядра Linux (Linux Kernel):

Android заснований на ОС Linux версії 2.6, тим самим платформі доступні системні служби ядра, такі як управління пам'яттю і процесами, забезпечення безпеки, робота з мережею і драйверами. Також ядро служить шаром абстракції між апаратним та програмним забезпеченням.

2.2 Apple IOS

Щоб ваш мобільний помічник міг виконувати свої завдання одного заліза й батареї недостатньо. Потрібна ще програмна начинка, яка забезпечить потрібний функціонал пристрою, тобто операційна система (ОС) та додатки (додатки). Так які ж мобільні ОС бувають і чим один від одного відрізняються?



Рис. 2.2. iOS 6 проти iOS 7

Напевно в світі залишилося зовсім мало людей, які не чули про плеєрах, телефонах iPhone та інших чудових продуктах яблучної компанії, а більш допитливі розуми напевно чули і про IOS, операційну систему Apple, під якою працюють такі її продукти, як iPod Touch, iPhone і Ipad.

- Інтерфейс:

У момент появи першого iPhone інтерфейс був досить інноваційним для смартфона. Великі іконки на які легко і зручно натискати пальцем. Велика клавіатура, також зручними для пальців (пальцелюбівая). Кілька віртуальних робочих столів з іконками для організації додатків (додатків). Нічого зайвого. Тільки сторінки з іконками, ніяких вам меню або діалогових вікон.



Рис. 2.3. Інтерфейс iPhone

До речі, таке цікаве явище як док (док), віртуальна полиця, куди ставляться найбільш затребувані додатки, теж вперше було масово поширено в продуктах Apple,. Як бачите, в інтерфейсі IOS він теж є і може зберігати до 4 іконок.

Інтерфейс IOS стандартизований і не змінювався від версії до версії. Налаштувати його за своїм смаком, вам також навряд чи вдасться. Ви мало що можете змінити, хіба що зробивши втечу з в'язниці, про який мова піде нижче.

- Архітектура IOS:

IOS спроектована таким чином, щоб, по-перше, зробити систему стабільною, по-друге, знизити витрату електрики і продовжити життя

акумулятора. Навіть якщо це означає, що доведеться відмовити від частини функціоналу і красивостей.

Кожна програма в IOS запускається в "пісочниці" (пісочниця), що виключає можливість впливу однієї програми на інші, як на рівні файлової системи, так і на рівні оперативної пам'яті. На практиці це означає, з одного боку, стабільність систем і те, що IOS не зависає, також якщо додаток App1 завершується некоректно (просто закривається саме по собі або чудить), це ніяк не вплине на роботу додатків App2 і App3 .

Ізоляція додатків сильно підвищує безпеку IOS, оскільки додатки можуть звертатися лише до файлів у своїй "пісочниці", так що ваші замітки ніяк не зможуть бути таємно або випадково відправлені, скажімо, в блог через додаток Livejournal App.

Однак, не обходиться без незручностей. Наприклад, якщо ви завантажили ролик з мережі через спеціальну "гойдалку" ви не зможете переглянути його, якщо в "гойдалці" ні функціоналу відео-плеєра. І перекинути файли з одного додатка в інше немає ніякої можливості.

Ще одна цікава особливість, цей поділ ресурсів. Запущеному додатку в IOS віддаються всі доступні ресурси, а всі інші програми закриваються. Ось чому таке явище як "гальма" можна вкрай рідко зустріти на пристроях Apple. Більш того, в стані коли всі програми закриті, можна бути впевненим, що " в тлі" не залишилося якогось сервісу, який з'їсть всю батарею або викачає весь ліміт мобільного трафіку, поки ви обідаєте або спите.

Хоча стабільність та енергозбереження це дуже важливо, але в ряді випадків однозначність зводить нанівець всі плюси такого підходу. Уявіть собі, що ви спілкуєтеся по скайп-чату або через інший месенджер і вам присилають щось на кшталт "Привіт, я там прислав тобі на пошту презентацію...". На жаль, для того, щоб ви змогли дістатися до пошти вам доведеться закрити Skype, і відкрити пошту, потім все спочатку, але у зворотний бік. Точно також можна " докачувати" ролик з YouTube паралельно читаючи книгу або відправляючи твіт.

Втім, буде несправедливо не сказати про деякі виправдовують ситуацію моментах. Майже у всіх пристойних програмах для спілкування вбудовані браузер, що дозволяють швидко переглянути надіслану посилання, це перше. Друге, майже всі програми, які написані в самій Apple, (браузер, будильник, пошта, замітки ...) можуть бути відкриті повторно і ви знайдете їх в тому вигляді, в якому залишили. І, зрозуміло, музику можна слухати завжди. Додаток ставку працює паралельно з будь-якими іншими, не використовують введення / висновки звуку (типу Skype).

У IOS 7 з'явилася-таки багатозадачність, тобто можливість запускати кілька додатків паралельно, однак вона працює не з усіма додатками і не завжди так, як треба.

Остання архітектурна особливість IOS це деякі непорозуміння. Так, не дивлячись на відмінну залізну начинку, IOS не підтримує передачу файлів через Bluetooth. Також не можна використовувати Bluetooth-гарнітуру для пристроїв iPod Touch, хоча є можливість приєднати провідний мікрофон. І того абсурдними, якщо підключити бездротові навушники, то працюють тільки кнопка старт / пауза, а тому / вперед такі дивацтва кілька збивають з пантелику.

- Програми IOS:

Для пошуку і установки додатків у Apple, є спеціальний сервіс, який називається AppStore (магазин додатків , якщо дослівно). Додатки бувають платними (від 0,99 до 900 \$) і безкоштовними. Всі додатки, перш ніж потрапити в AppStore проходять перевірку на ідеологічну складову і нешкідливість для МО. Така санітарна обробка дозволяє уникнути шкідництва з боку розробників і захистити особисті дані (і гроші) користувачів від крадіжки.

Більшість додатків в AppStore не представляють інтересу і часто дублюють один одного. Одних тільки будильників можна знайти з півсотні, не кажучи вже про календарях, тижневиках і записних книжках. Однак, серед них є маса корисних, які допоможуть вам організувати свої справи,

підкажуть, направлять і розважать, та й взагалі, бродити по AppStore майже також цікаво, як по справжньому магазину.

Крім AppStore також існує магазин Cydia. Cydia можуть користуватися ті, хто попередньо розблокував свій телефон через процедуру джейлбрейка. У Cydia можна знайти масу корисних додатків (як платних, так і безкоштовних), які не змогли потрапити в AppStore або ніколи туди не прагнули. Оскільки Apple, фільтрує додатки вельми прискіпливо, багато з програм змушені розміщуватися через цей альтернативний магазин. Втім, не буду псувати вам задоволення від дослідницької діяльності.

- Jailbreak и Unlock:

Не всі люди готові мириться з обмеженнями, які Apple, накладає на їхні пристрої. Зокрема при покупці телефону iPhone, ви, швидше за все, придбаєте його у оператора на умовах контракту або у перекупника, який продасть вам телефон, який був до цього придбаний через оператора.

Проблема полягає в тому, що за умовами поставки iPhone оператор зобов'язаний укласти з клієнтом дворічний контракт за яким клієнту слід виплачувати певну суму щомісяця за деяке число розмовних хвилин і мобільного інтернету. Більше того, ви не зможете перейти до іншого оператора зберігши за собою телефон, поки не закінчитися термін дії договору. Те ж правило поширюється і на роумінг. Коротше кажучи ви не можете використовувати телефон інакше як з сім-картою оператора, у якого придбали iPhone.

Втеча з в'язниці (втеча з в'язниці, дослівно) це процедура, що дозволяє отримати повний контроль над IOS, тобто встановлювати будь-які додатки, переглядати всю файлову систему, змінювати системні налаштування, а також активувати багатозадачність на iPhone 3G і iPod Touch 2-го покоління і багато іншого.

Однак, слід врахувати, що процедура суттєво впливає на стабільність пристрою і знижує рівень безпеки, крім того, акумулятор може почати розряджатися швидше.

- Взаємодія з ПК:

Пристрої під керуванням IOS 4 можуть працювати як з Mac'ами, так і з вікнами або Linux. Грустність полягає в тому, що якщо в Mac'ах iTunes - це стандартний додаток, то в Windows, вам доведеться завантажити близько 100Мб софта, щоб просто закачати музику в плеєр. Те ж саме стосується й інших операцій, таких як резервне копіювання, оновлення прошивки, синхронізація контактів і т.д. Без iTunes ви не зможете активувати свій iPhone або зареєструватися для роботи з AppStore. До речі, ваші налаштування і музика можуть бути синхронізовані тільки з ОДНИМ пристроєм, також як ваш пристрій може бути синхронізоване для роботи тільки з одним ПК.

Загалом, звучить не дуже, правда? Насправді з цим цілком можна жити, головне звикнути. З Linux справи йдуть трохи краще, як не дивно, ви можете працювати зі своєю музичною колекцією без всяких проблем через такі програми як Rhythmbox (стандартний плеєр Ubuntu), свої календарі та контакти можна синхронізувати через акаунт у Google, а додатки встановлювати через мобільний інтерфейс AppStore.

2.3 Windows Phone

Windows Phone - мобільна операційна система, розроблена Microsoft, вийшла 11 жовтня 2010. 21 жовтня почалися поставки перших пристроїв на базі нової платформи. У Росії телефони з Windows Phone почали продаватися 16 вересня 2011, першим з яких став HTC 7 Mozart.



Рис. 2.4. Логотип Windows Phone

Операційна система є наступником Windows Mobile, хоча і несумісна з нею, з повністю новим інтерфейсом і - вперше - з інтеграцією сервісів Microsoft: ігрового Xbox Live і медіаплеєра Zune. Презентація системи відбулася в рамках заходу Mobile World Congress 2010, що пройшов у Барселоні.

Windows Phone підтримує технологію мультитач і має новий користувальницький інтерфейс під назвою Metro, принципи якого були раніше використані в дизайні інтерфейсу Windows Media Center, Zune і Xbox. Також в Windows Phone знайшли відображення ідеї платформи смартфонів Microsoft Kin.

Microsoft переробила початковий екран: тут більше немає статичних іконок - всі вони замінені на так звані «живі плитки» (англ. Жива плитка) (або «іконки», «тайли»), які відображають інформацію в режимі реального часу без участі користувача. «Живі плитки» прокручуються по вертикалі і можуть служити як звичайним ярликом для програми, так і ярликом для контакту, замітки, веб-сторінки, медіаконтенту та іншого. Більшість «плиток» мають квадратну форму, але поряд з ними інтерфейс операційної системи містить і прямокутні (фото, календар).

Одне з нововведень інтерфейсу. «Хаби» (англ. хаби) або, як описано на офіційному сайті, «розділи». Всього в системі предумовлено 6 таких розділів, але виробники телефонів можуть створювати свої, наприклад, HTC Hub, відповідно від компанії HTC. Розділ «Контакти» (англ. People Hub) об'єднує всю інформацію, що стосується якогось певного людини, в тому числі його записи та коментарі в соціальних мережах, а також фотографії, надаючи централізований доступ до таких мереж, як facebook, Twitter, LinkedIn і Windows Live. З Windows Phone 8 в розділі «Контакти» з'явилася можливість створювати «кімнати»: список певних контактів, яким доступна та чи інша інформація. Розділ «Фото» (англ. Фотографії концентратор) об'єднує фотографії користувача, що зберігаються в пам'яті пристрою, на комп'ютері та в інтернеті. За допомогою цього розділу можна опублікувати фотографію в соціальних мережах, помітити на ній людини, а також мати доступ до останніх фотографій від друзів. Розділ «Ігри» (англ. Ігри концентратор) відкриває доступ до ігор і сервісу Xbox Live, де можна подивитися свої чи чужі профілі, аватари, досягнення. Розділ «Музика + відео» (англ. Музика + Відео концентратор) об'єднує весь мультимедійний

контент, що зберігається на комп'ютері користувача, музичні онлайн-сервіси, вбудоване FM-радіо і відкриває доступ до сервісів Xbox Музика і Xbox Video. Розділ «Магазин» дозволяє купувати або завантажувати безкоштовно програми та ігри, а розділ Управління забезпечує доступ до мобільної версії Офіс-Office Mobile, включаючи Word, Excel, OneNote, PowerPoint. За допомогою офісного пакету користувач має можливість відкрити, створити і редагувати документи, а потім синхронізувати зроблені зміни з настільним комп'ютером, використовуючи сервіс OneDrive.

- Windows Phone 7:

15 лютого 2010 в Барселоні на Mobile World Congress 2010 Стівом Балмером вперше була анонсована нова мобільна операційна система - Windows Phone 7. Офіційно була представлена 11 жовтня 2010, а 21 жовтня о продажу з'явилися перші смартфони на новій платформі в Європі (Великобританії, Німеччини, Франції, Іспанії та Італії) і Азіатсько - Тихоокеанському регіоні (Сінгапурі, Австралії та Нової Зеландії). Windows Phone 7 була доступна на п'яти мовах.

- Windows Phone 7.5:

У лютому 2011 року на Mobile World Congress 2011 корпорація Microsoft вперше анонсувала нове оновлення Windows Phone, а вже у квітні на конференції MIX'2011 в Лас-Вегасі компанія розповіла про подробиці цього оновлення, яке отримала назву Mango. Пізніше компанія офіційно заявила, що це оновлення операційної системи отримає порядковий номер 7.5 у новій версії, яка вийшла остаточно у вересні 2011 року, більше 500 поліпшень і доповнень, що зробило роботу операційної системи швидше у системі оновився браузер Internet Explorer Mobile до версії 9, який тепер підтримує HTML5, Canvas, апаратне прискорення. З'явилася можливість використовувати фронтальну камеру. Одне з найважливіших змін для російськомовних користувачів - це локалізація на російську мову - інтерфейс, клавіатура, сумісність з кирилицею. У загальній складності тепер платформа підтримує 24 мови.

У лютому 2012 року на Mobile World Congress 2012 був продемонстрований смартфон Nokia Lumia 610 з черговим оновленням операційної системи, що є частиною Манго, в якому з'явилася підтримка процесорів з частотою 800 МГц і оперативною пам'яттю в 256 Мб. Особливість даного оновлення полягає у можливості випуску на ринок смартфонів бюджетного сегмента. У березні це оновлення було офіційно названо як Windows Phone 7.5 Refresh .

- Windows Phone 7.8:

В кінці 2012 року в офіційному блозі Windows Phone розробники повідомили про заплановане випуску на початку 2013 року поновлення під номером 7.8, яка включатиме в себе деякі нововведення платформи Windows Phone 8: стартовий екран з налаштованим розміром «живих плиток », новий екран блокування , створення рингтонів, передача даних через Bluetooth. 30 січня 2013 компанія Nokia офіційно оголосила про початок оновлення своїх апаратів лінійки Lumia до версії 7.8, яке містить обіцяні зміни.

Повноцінного поновлення смартфонів на Windows Phone 7.x до версії 8 не буде, що офіційно підтвердила Microsoft. Офіційна підтримка платформи Windows Phone 7 припиниться через півтора року після виходу оновлення під номером 7.8, тобто приблизно в середині 2014 року. Версія 7.8 не тільки оновить поточні апарати на Windows Phone 7/7.5, але і буде встановлюватися на бюджетні WP -пристрої.



Рис. 2.5. Логотип Windows Phone 8

Windows Phone 8

20 червня 2012 на організованій Microsoft конференції під назвою Windows Phone Summit була анонсована Windows Phone 8. Головна перевага Windows Phone 8 . Можливість об'єднати планшети, смартфони та персональні комп'ютери в єдину «екосистему», тобто можливість створення

умов для розробників, полегшують портирование програмного забезпечення між цими пристроями. Нове покоління платформи буде засновано на ядрі NT і максимальний дозвіл екрану апаратів буде 1280×768 або 1280×720, на відміну від Windows Phone 7.x, яка заснована на ядрі CE з дозволом екрану 800×480.

29 жовтня 2012 Microsoft офіційно представила наступне покоління платформи. Нововведення Windows Phone 8. Internet Explorer Mobile 10 -й версії з функцією виявлення шкідливих сайтів, підтримка багатоядерних процесорів, технологія обміну даними NFC, слот для карт пам'яті та інші. Користувачі отримали можливість змінювати розмір «живих плиток» стартового екрана. тепер їх три види (маленькі, середні і великі).

- Windows Phone 8.1:

Компанія Microsoft 2 квітня 2014 на конференції BUILD показала версію системи під номером 8.1. Серед нововведень оновленої системи: центр повідомлень, голосовий помічник під ім'ям Кортан, нова клавіатура Слово потоку, підтримка корпоративних VPN.

- Магазин додатків:

Для пристроїв на Windows Phone передбачений інтернет-магазин програм та ігор Windows Phone Marketplace , доступний в 60 країнах. Купівля або установка цих програм можлива через розділ Marketplace на телефоні або через браузер. В кінці червня 2012 року Microsoft офіційно заявила, що в Windows Phone Marketplace кількість додатків перевищило за 100 тисяч.

Для покупки музики, відео, подкастів, телевізійних передач існує інтернет -магазин Zune Marketplace, який доступний на телефоні через розділ «Музика + відео» або через спеціальну програму встановлену на комп'ютері з операційною системою Windows - Zune Software. Zune Software - це повноцінний програвач, каталогізатор, який синхронізує пристрій і персональний комп'ютер, а також використовується для оновлення операційної системи смартфонів. Для Mac OS X компанія Microsoft випустила аналогічний додаток - Windows Phone 7 Connector, яке, крім

іншого, дозволяє синхронізувати Windows Phone з частиною бібліотеки iTunes (фільми, музику, аудіо і відео подкасти) і iPhoto.

Встановлення програм та ігор на Windows Phone можлива тільки з офіційного інтернет -магазину Windows Phone Marketplace . Для ентузіастів можливе розблокування телефону за допомогою затвердженого Microsoft сервісу ChevronWP Labs, в результаті чого на смартфон можна встановлювати саморобні програми або використовувати його для тестування в обхід Marketplace та офіційної розблокування телефону в якості розробника.

У вересні 2012 року Microsoft офіційно перейменувала Windows Phone Marketplace в Windows Phone Store, оновивши і поліпшивши пошук за додатками. З 30 листопада, після чергового оновлення Магазину Windows Phone, для пристроїв на платформі Windows Phone 8 з'явилася можливість встановлювати програми та ігри з SD -карти вручну.

Для розробки додатків і ігор використовується Silverlight або XNA. Microsoft випустила інструментарій розробника Windows Phone SDK , для якого необхідні Visual Studio 2010 Express for Windows Phone, Expression Blend 4 for Windows Phone, XNA Game Studio 4.0. Формат файлів установки змінився с. Cab, використовуваний в Windows Mobile, на новий - хар.

В кінці 2012 року Microsoft офіційно повідомила про доступність більше 150 тисяч додатків в Windows Phone Store і про функціонування магазину в 191 країні. Влітку 2013 року кількість додатків в магазині досягла 165 тисяч. У серпні цього ж року кількість завантажень в Windows Phone Store досягло 2 мільярди.



Рис. 2.6. Windows Phone Store

Windows Phone SDK

- Розробка під Windows Phone 7:

Всі програми для Windows Phone 7 створюються з використанням керованого коду. NET. В даний час C # - єдиний підтримуваний мову програмування. Вільно доступний для завантаження Microsoft Visual Studio 2010 Express для Windows Phone включає XNA Game Studio 4.0 і екранний емулятор телефону, а також інтегрується з Visual Studio 2010.

Візуальні елементи і анімація для додатків Silverlight можуть створюватися у Microsoft Expression Blend . Платформи Silverlight і XNA для Windows Phone 7 мають ряд загальних бібліотек, тобто деякі бібліотеки XNA можуть використовуватися в програмі Silverlight і навпаки. Але не можна створювати програму, що поєднує в собі візуальні елементи обох платформ.

Як правило, Silverlight використовується для програм, які можна класифікувати як додатки або утиліти. Опис компоновки елементів управління і панелей користувацького інтерфейсу в цих програмах виконується за допомогою розширювана мова розмітки додатків (Extensible Application Markup Language, XAML). У файлах виділеного коду можуть реалізовуватися операції з ініціалізації і деяка логіка, але основним їх призначенням є обробка подій елементів управління. Silverlight дозволяє реалізовувати в Windows Phone стиль насичених інтернет-додатків (Rich Internet Applications, RIA), включаючи мультимедіа та веб. Для Windows

Phone створена версія Silverlight 3, в яку не ввійшли деякі можливості, що не підходять для телефону, але компенсовані поруч доповнень.

Головне призначення XNA - створення високопродуктивних ігор. Для 2D-ігор спрайт і підкладки описуються за допомогою растрових зображень; для 3D ігор створюються тривимірні моделі. Дія гри, що включає переміщення графічних об'єктів по екрану і запит користувача введення, обробляється вбудованим ігровим циклом XNA.

Зручно провести кордони і прийняти, що Silverlight використовується для додатків, а XNA - для ігор, але це не повинно накладати обмеження. Поза всяких сумнівів, Silverlight може застосовуватися для реалізації ігор, і традиційні додатки можуть створюватися на XNA, хоча це буде пов'язане зі значними труднощами.

Silverlight підходить для ігор з невеликими вимогами з графіки, або використовують векторну, а не растрову графіку, або темп яких визначається реакцією користувача, а не таймером.

III. РОЗРОБКА МОБІЛЬНОГО МУЛЬТИ-МЕДІА ДОДАТКУ НА БАЗІ ПЛАТФОРМИ ANDROID

У додатках під Android використовуються бази даних SQLite, які являють собою один з п'яти способів зберігання даних в Android-. Ми будемо розглядати тільки бази даних SQLite, оскільки це і є основа побудови робочої і функціональної програми. Після освоєння цього посту ви зможете вставляти дані з таблиць в бази даних, і проводити їх відбір.

Почнемо зі створення нового проекту під назвою PROSTO-FM. У першій частині нашої серії статей про Android-розробках процес створення нового проекту був детально описаний, так що зупинятися на покроковій інструкції знову ми не будемо. Дані, які необхідно внести в форму створення нового проекту наступні:

Назва проекту: TEST

Побудувати об'єкт: Android 1.5

Ім'я програми: TEST

Ім'я пакету: com.gregjacobs.randomquotes

Створення активність: QuotesMain

Мін SDK Версія: 3

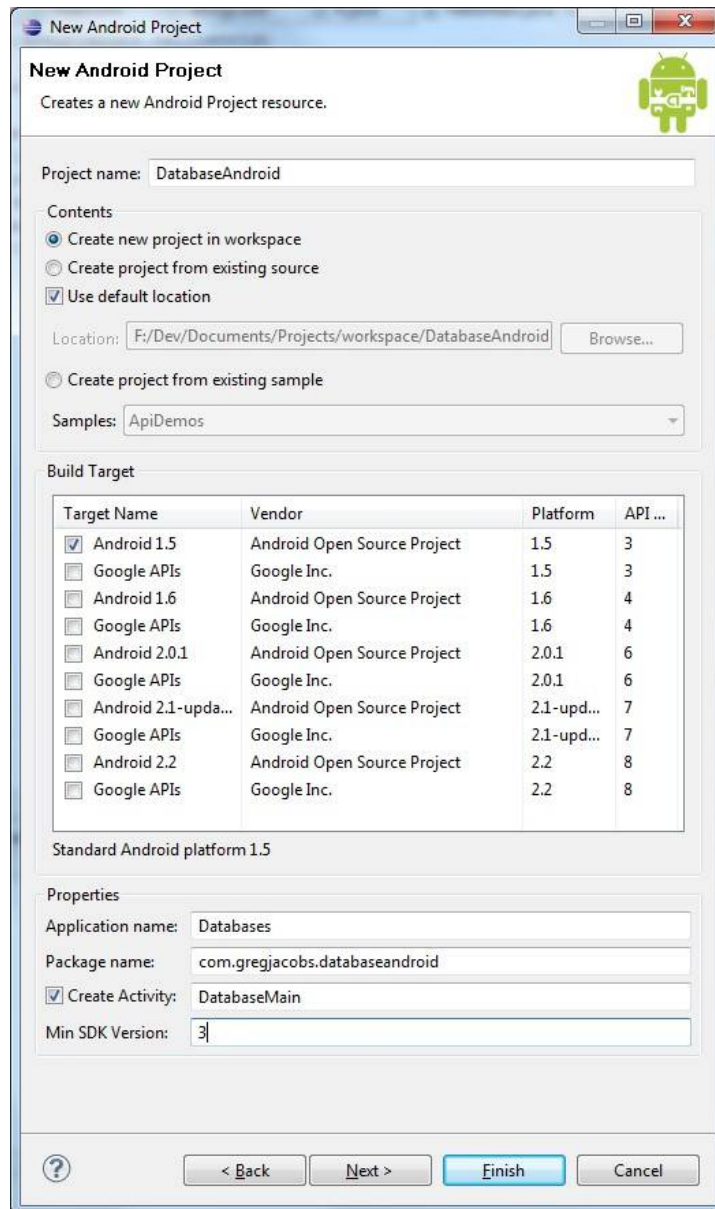


Рис. 3.1. Створення нового проекту

Після введення даних і натискання на кнопку Готово приступимо до створення файлу класу в нашому пакеті під назвою `com.gregjacobs.randomquotes`. Для цього потрібно натиснути правою кнопкою миші на пакеті, вибрати в меню Новий, потім клас. У вікні потрібно заповнити лише поле Ім'я, ввівши в нього `DBAdapter`. Далі тиснемо Finish і отримуємо базовий файл для класу, який нам належить трохи видозмінити. У цьому розділі я буду діяти також, як і в попередньому: спочатку наведу код, а потім поясню основні його частини, призначення головних функцій. В додаток в цей раз я приведу текстові файли, так щоб ви могли завантажити і порівняти зі своїм варіантом. Почнемо з файлу `DBAdapter.java`:

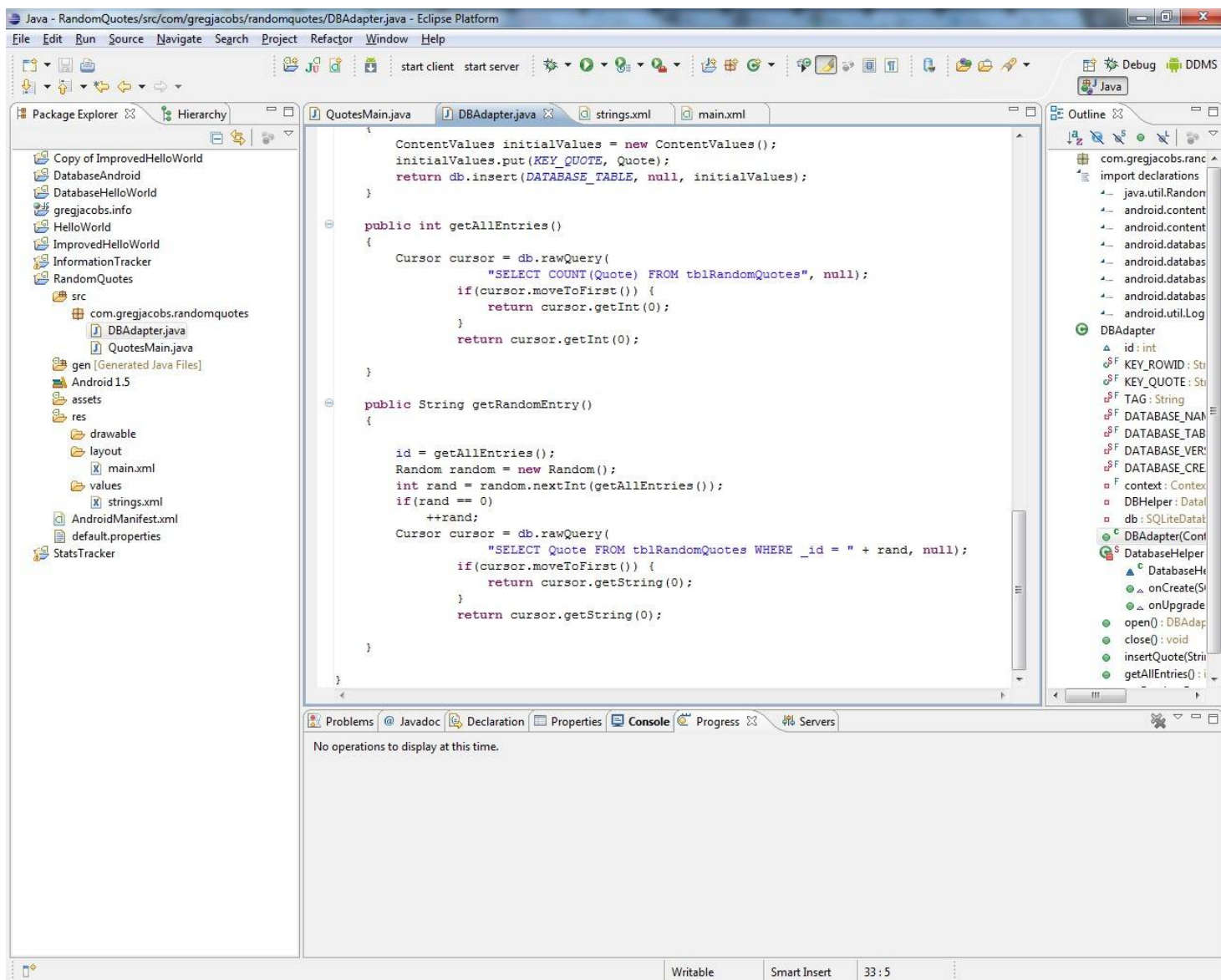


Рис.3.2. Вигляд головного вікна програми

Почнемо з імпорту всіх інструментів, які знадобляться для створення і функціонування нашої бази даних SQLite. Можливо, професійним програмістам не знадобляться подальші пояснення, що означає кожен рядок, але для початківців вони будуть корисними. Отже, ContentValues дає можливість зберігати набір значень для операторів вставити (оператор вставки), контекст, як вже було сказано в попередньому пості, дає доступ до середовища додатки. Курсор-напевно, самий потрібний імпорт з усіх. Курсор дозволяє мати доступ до даних, отриманих з БД. SQLException дозволяє викидати SQL виключення при появі помилки. Ці повідомлення дозволяють зрозуміти, чим конкретно викликана помилка. SQLiteDatabase дає можливість керувати базою даних SQLite, використовуючи методи.

SQLiteOpenHelper являє собою клас-помічник в управлінні БД. Ввійти буде Залогуватися виведення даних у разі виникнення помилки.

```
« public class DBAdapter
{
    int id = 0;
    public static final String KEY_ROWID = "_id";
    public static final String KEY_QUOTE = "Quote";
    private static final String TAG = "DBAdapter";

    private static final String DATABASE_NAME = "Random";
    private static final String DATABASE_TABLE = "tblRandomQuotes";
    private static final int DATABASE_VERSION = 1;

    private static final String DATABASE_CREATE =
        "create table tblRandomQuotes (_id integer primary key autoincrement, "
        + "Quote text not null );";

    private final Context context;

    private DatabaseHelper DBHelper;
    private SQLiteDatabase db; »
```

Лістинг 1. Код програми для визначення використовуваних змінних

Тут ми визначаємо всі потрібні нам змінні, починаючи з назви бази даних і закінчуючи запитом створення самої бази даних. Ми використовуємо остаточне змінні т.к. їх значення не будуть змінюватися в ході роботи програми, але при цьому, виносимо змінні начебто імен таблиці в окремі змінні, щоб полегшити модифікацію програми надалі.

```
«
public DBAdapter(Context ctx)
{
    this.context = ctx;
    DBHelper = new DatabaseHelper(context);
}
private static class DatabaseHelper extends SQLiteOpenHelper
{
    DatabaseHelper(Context context)
    {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
    @Override
    public void onCreate(SQLiteDatabase db)
    {
        db.execSQL(DATABASE_CREATE);
    }
}
```

```

    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion,
        int newVersion)
    {
        Log.w(TAG, "Upgrading database from version " + oldVersion
            + " to "
            + newVersion + ", which will destroy all old data");
        db.execSQL("DROP TABLE IF EXISTS tblRandomQuotes");
        onCreate(db); }
    }
    »

```

Лістинг 3.2. Код програми для визначення конструкторів

У вищевказаному фрагменті коду ми визначаємо конструктор, який буде передавати контекст пропозиції нашому помічникові, DatabaseHelper. Клас DatabaseHelper розширює можливості нашого SQLiteOpenHelper, який покращує функціонал управління базою даних SQLite. Головна функція onCreate, яку нам ще належить використовувати далі, дозволяє виконати SQL-запит по створенню бази даних.

```

public DBAdapter open() throws SQLException
{
    db = DBHelper.getWritableDatabase();
    return this;
}

public void close()
{
    DBHelper.close();
}

```

Лістинг 3.3. Функції відкриття та закриття бази даних

У коді вище є дві ключових функції відкриття і закриття бази даних. На ці функції можна посилатися при виклику їх у нашому головному. Java файлі.

```

public long insertQuote(String Quote)
{
    ContentValues initialValues = new ContentValues();
    initialValues.put(KEY_QUOTE, Quote);
    return db.insert(DATABASE_TABLE, null, initialValues);
}

```

Лістинг 3.4. Функція для додавання цитат до бази даних

Вище описана функція обробляє наші цитати, коли ми викликаємо їх в головному. Java файлі. Також ця функція готує цитати для введення в БД, поміщаючи рядок Цитата в ContentValues під назвою initialValues, які потім вставляються в таблицю БД.

```
public int getAllEntries()
{
    Cursor cursor = db.rawQuery(
        "SELECT COUNT(Quote) FROM tblRandomQuotes", null);
    if(cursor.moveToFirst()) {
        return cursor.getInt(0);
    }
    return cursor.getInt(0);
}
```

Лістинг 3.5. Функція запиту

Ця функція буде виконувати запит на кількість введених в БД цитат для передачі цього значення як максимально можливого генератору випадкових чисел. Таким вдасться уникнути помилки (обраний номер цитати ніколи не буде перевищувати максимально можливий) . Ми в основному використовуємо в роботі тип запиту rawQuery , бо особисто мені не дуже подобається як Android обробляє свої запити , але я вражений можливістю використання повнофункціональних SQL -запитів. Умова «якщо» змусить показчик прийняти перший результат (якщо буде знайдено кілька результатів). Якщо умова "якщо" не дійсно , все одно за результат прийметься значення першої позиції.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/Quote"
    />
    <EditText
        android:id="@+id/Quote"
        android:layout_width="fill_parent"
```

```
        android:layout_height="wrap_content"
    />
    <Button
        android:id="@+id/go"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/press"
    />
    <Button
        android:id="@+id/genRan"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/genRan"
    />
</LinearLayout>
```

Лістинг 3.6. Приклад коду для візуального оформлення першої версії програми

4. РОЗРОБКА ДОДАТКІВ ДЛЯ МОБІЛЬНИХ ПЛАТФОРМ IPHONE, ANDROID, BLACKBERRY

4.1. Вступ

Останні кілька років у світі спостерігається все більше поширення пристроїв на базі мобільних платформ. З'являються нові мобільні платформи. У цей час основна частина ринку поділена між декількома мобільними платформами. З погляду різних аналітичних агенств, найбільше активно серед них у найближчі кілька років будуть розвиватися iPhone, Android і Blackberry.

По своїй продуктивності мобільні пристрої вже досягли рівня настільних комп'ютерів. Раніше до платформ, під керуванням яких працювали, наприклад, мобільні телефони, пред'являлися тверді вимоги. Невід'ємною частиною мобільних платформ був набір додатків, ретельно налагоджених, протестованих виробником і завантажених на мобільний пристрій разом із самою платформою. Не було широко поширене завантаження сторонніх додатків. Для нових типів мобільних платформ, навпаки, завантаження сторонніх додатків є однією з основних можливостей. Сторонні розроблювачі одержали можливість створення й установки своїх додатків на мобільні платформи. У той же час компанією Apple був створений новий механізм масового поширення мобільних додатків - App Store. Механізм виявився успішним і був узятий на озброєння іншими компаніями-розроблювачами мобільних платформ. Цей механізм дозволив стороннім розроблювачам діставати прибуток від створення й поширення мобільних додатків.

Зараз у сегменті ринку мобільних додатків працюють окремі розроблювачі й групи розроблювачів, безпосередньо орієнтовані на розробку мобільних додатків, а також ІТ-компанії з багаторічним досвідом розробки програмного забезпечення, які відкрили новий напрямок своєї діяльності.

Розробка мобільних додатків у багатьох аспектах подібна з розробкою додатків для настільних комп'ютерів. У той же час розробка мобільних додатків ставить перед програмістами нові завдання. До таких завдань, зокрема, ставляться більш уважне керування пам'яттю мобільної платформи, облік обмеженого заряду батареї, а також облік можливої нестабільності бездротового мережного з'єднання.

Одержання знань по розробці додатків для мобільних платформ може бути цікаво як з погляду застосування в професійній діяльності, так і з погляду більш глибокого розуміння роботи мобільного програмно-апаратного комплексу.

Мобільні пристрої на базі нових мобільних платформ, у тому числі смартфони, стають частиною повсякденного життя. Вони використовуються для спілкування. Поряд зі стандартними можливостями мобільних телефонів, як-от: дзвінки, обмін текстовими повідомленнями через SMS або в соціальній мережі, - створюються додатки, що інтегрують функції обміну інформацією, що працюють одночасно з декількома соціальними мережами. Усе активніше відбувається обмін відеоінформацією на основі, що вбудовуються в смартфони камер. У подорожах виявляється дуже затребувана функціональність GrstНавігатора. Роста інтерес до інтеграції інформації від GrstНавігатора смартфона в соціальні мережі. У розроблювачів з'являється можливість використовувати власні додатки у своєму повсякденнім житті, що дозволяє їм по-новому глянути на свій продукт. При навчанні розробці мобільних додатків цей фактор може додатково мотивувати й збільшити інтерес до процесу навчання.

4.2. Розробка мобільних додатків

Однієї з перших тенденцій, що супроводжують швидкий розвиток мобільних платформ, став перенос на них практично всіх технологій, розвинених для настільних комп'ютерів [1]. У розробці додатків намітилися кілька напрямків. Серед них варто відзначити ігрові додатки, додатки для роботи із соціальними мережами й бізнесдодатки, орієнтовані на документообіг у середніх і великих компаніях. По способу розробки виділилися два основні види додатків: "native" додатка й web-додатка. "Native" додатки створюються повністю на базі власного API мобільної платформи. Webдодатка включають клієнтський додаток, на основі невеликої кількості "native" коду, що одержує контент від вилученого серверного додатка, що й відображає його з використанням стандартних webтехнологій, у тому числі HTML, Java Script. Вилучений серверний додаток може бути створене заново або може бути задіяний наявний серверний додаток з відомим API.

Ігрові додатки пишуться на "native" коді для досягнення максимальної продуктивності. Для додатків, що працюють із соціальними мережами, створюється клієнтський додаток, що використовує для одержання даних

стандартний API сервера тієї або іншої соціальної мережі. Для розробки бізнесдодатків використовується як "native" код, так і web-технології.

При розробці мобільного додатка програміст повинен розуміти, як його додаток використовує пам'ять платформи. Кілька корисних підходів для зменшення використання пам'яті при розробці мобільних додатків представлені в статті "On Mobile Java Memory Consumption" [2]. Зазначені підходи в першу чергу орієнтовані на платформу Java ME. Однак вони можуть бути поширені й на інші мобільні платформи. Підходи включають відмова по можливості від використання маленьких класів в одному коді, тому що класи вимагають додаткової пам'яті, і, коли необхідний окремий оброблювач команд для елемента користувацького інтерфейсу, рекомендується використовувати основний об'єкт як оброблювача, навіть якщо потрібна додаткова логіка. Інший підхід полягає в тому, щоб уникати залежностей між класами й підбирати розмір таких структур даних, як вектори й рядка. Необхідно приділяти увагу спадкуванню й переконуватися, що будь-який клас, який успадковує методи, використовує ці методи. Усі ці підходи сприяють збільшенню продуктивності мобільного додатка. Їхнє застосування буде сприяти формуванню тих, яких навчають, як більш компетентних програмістів.

Іншою проблемою для мобільних пристроїв є велика різноманітність їх екранів і дозволів. Практичне рішення для створення "гнучкого" графічного користувацького інтерфейсу для платформи Java ME показане в статті "Better Midlets by design" [3]. Воно засноване на приділенні уваги тому, як пункт меню відображається на екрані при використанні високорівневого API платформи Java ME, а у випадку використання низкорівневого API вимагає більшого часу тестування на різних пристроях.

4.3. Мобільні платформи

У даний момент серед досить широкої різноманітності мобільних платформ на світовому ринку виділяються iPhone, Android і Blackberry. Платформи iPhone і Blackberry розробляються й підтримуються на ринку в основному тільки компаніями-розроблювачами. Вихідні коди цих платформ закриті. На противагу iPhone і Blackberry, платформа Android, розроблена компанією Google, підтримується альянсом компаній ОНА (Open Handset Alliance), серед яких присутні такі компанії, як HTC, Intel, Samsung. Вихідні коди цієї платформи відкриті. Це дозволяє різним компаніям-розроблювачам мобільних апаратних систем використовувати Android у якості базової

програмної платформи. Кожна компанія може доробити платформу Android у відповідності зі своїми вимогами.

Кожна з перерахованих вище мобільна платформа має свою програмну мову й інструментарій для розробки додатків. Важливим питанням для програміста, що розв'язав спробувати свої сили в розробці мобільних додатків, є вибір мобільної платформи. І дійсно, програмна мова й інструментарій для платформ iPhone, Android і Blackberry суттєво відрізняються. Перехід між ними багато в чому вимагає повторного вивчення нової платформи.

4. 4. Мови й інструментальні засоби

4.4.1 iPhone

Використовується мова Objective-C. Розробка ведеться на настільній операційній системі Mac OS X. У якості інтегрованого середовища розробки використовується Xcode. З Xcode інтегровані додаток для пошуку витоків і спостереження за виділенням пам'яті Instruments і додаток Interface Builder для створення графічного інтерфейсу. Для симуляції роботи додатка на мобільному пристрої служить iPhone Simulator. З використанням iPhone Simulator не можна перевірити роботу деяких специфічних API, наприклад роботу з акселерометром, геопозиціонуванням і камерою. iPhone Simulator входить до складу iPhone SDK (Software Development Kit). iPhone SDK містить фреймворки й бібліотеки для складання додатків під реальний пристрій iPhone і iPhone Simulator. Xcode і iPhone SDK можна завантажити on-line [4] з офіційного сайту розроблювачів додатків для платформ компанії Apple. Для завантаження інструментарію потрібна реєстрація. Реєстрація припускає участь в одній із програм для розроблювачів. Найбільш простий є програма iOS Developer Program. Вона безкоштовна й дозволяє завантажити й використовувати інструментарій розробки, а також завантажувати й тестувати власні додатки на iPhone Simulator. Можливість завантаження додатків на реальний пристрій відсутня. Для одержання безкоштовної можливості завантаження й тестування на реальному пристрої можна використовувати програму iOS University Program. Однак потрібне підтвердження того, що учасник програми дійсно буде використовувати реальний пристрій iPhone лише з метою університетського навчання.

На офіційному сайті розроблювачів додатків для платформ компанії Apple також можна знайти докладні інструкції з установки й використанню інструментарію.

4.4.2 Android

Споконвічно була можливість розробляти додатки тільки з використанням мови Java у формі JNI (Java Native Interface), розробленого компанією Google. Додатка для платформи Android збираються в байтткод для виконання віртуальною машиною Dalvik. При складанні додатків можливо використовувати сторонні жагафайли, зібрані для віртуальної машини Java. У процесі складання байтткод віртуальної машини Java перезбирається в байтткод віртуальної машини Dalvik спеціальною утилітою, що входить до складу інструментарію розробки додатків.

У червні 2009 вийшов перший офіційний реліз Android NDK (Native Development Kit). Розроблювачі одержали можливість розробляти власні бібліотеки мовою C для платформи Android. Використання коду мовою C у критичні по продуктивності ділянках додатка суттєво підвищує його швидкодія. Розробка може вестися на інструментальних машинах під керуванням Windows, Linux, Mac OS X. Існують два способи розробки: з використанням консольних утиліт і з використанням ADT plug-in для Eclipse. Для імітації роботи на реальному пристрої Android служить емулятор Android, створений на базі віртуальної машини Qemu. Емулятор Android входить до складу Android SDK. Також в Android SDK входять засоби профілювання, налагодження й створення графічного інтерфейсу додатків. Завантажити Android SDK і Android NDK можна on-line [5] з офіційного сайту розроблювачів для платформи Android. На тій же сторінці Internet розташоване посилання на покрокову інструкцію з установки інструментарію.

Використання інструментарію розробки під платформу Android, емулятора, а також завантаження й тестування на реальному пристрої не вимагають оплати.

4.4.3 Blackberry

Інструментальна машина для створення додатків може працювати під керуванням однієї з операційних систем: Windows, Linux, Mac OS X. Розробка додатків ведеться мовою Java. Використовується платформа Java

Micro Edition (Java ME) у конфігурації Connected Limited Device Configuration (CLDC) і Mobile Information Device Profile (MIDP). Платформа Java ME у зазначеній конфігурації використовується також для досить широкого набору інших платформ. Крім того, платформа Java ME простіше платформи Java Standard Edition (SE), вивчення якої часто входить у програму навчання комп'ютерним наукам. Це спрощує початок розробки мобільних додатків під платформу Blackberry для студентів факультетів комп'ютерних наук. Для ознайомлення з Java ME рекомендується використовувати Sun Microsystems Java Wireless Toolkit для CLDC. У неї вбудований емулятор роботи додатків на мобільних пристроях. Для розробки додатків як для Java ME, так і для Blackberry, використовується Blackberry Java Development Environment (JDE). JDE існує у вигляді окремого IDE і у вигляді plug-in для Eclipse. Blackberry JDE дозволяє використовувати такі функціональні можливості апаратної платформи Blackberry, як камера, медіа, акселерометр. JDE включає необхідні інструменти для конвертування існуючих Java ME Midlets у формат (.cod) для платформи Blackberry. Завантажити інструментарій для розробки додатків для платформи Blackberry можна після переходу по відповідних до посилань на сторінці [6] офіційного сайту платформи Blackberry.

5. Інструментарій кросм платформної розробки

Відмінності в мові й інструментарії розробки для розглянутих мобільних платформ визначають істотні тимчасові й економічні витрати при розробці й підтримці кросмплатформних додатків. А завдання створення мобільного додатка, здатного працювати на декількох платформах, залишається актуальною. Існує ряд кросмплатформних інструментів, які дозволяють використовувати той самий код для декількох мобільних платформ. Для складання додатків під конкретну платформу повинен бути встановлений SDK для цієї платформи. До кросмплатформним інструментам, приміром, ставляться Rhodes і Phonegap [7].

5.1 Rhodes

Інструмент створений компанією Rhomobile. Доступний у тому числі й для платформ iphone, Android і Blackberry. Дозволяє створювати кросмплатформні додатки з використанням HTML, CSS, Javascript і Ruby. Призначений для програмістів, які мають досвід webорозробки й прагнуть створювати мобільні додатки без необхідності вивчати інструментарій і мови для кожної мобільної платформи. Rhodes у першу чергу призначений для

створення бізнесдодатків з набором екранів, що полягають зі стандартних графічних елементів. Не призначений для розробки ігор, де потрібна висока продуктивність, а також додатків, що включають специфічні графічні елементи. Rhodes є відкритим продуктом.

5.2 Phonegap

Відкритий фреймворк для створення "native" додатків з використанням HTML, CSS і Javascript, у тому числі, для платформ iPhone, Android і Blackberry. Проект спонсорується компанією Nitobi. Дозволяє через Javascript Apis використовувати дані контактів, геопозиціонування, камери й акселерометра, які не доступні для web-додатків. Дозволяє легко конвертувати існуючі мобільні webдодатки для роботи в Phonegap. Phonegap Apis міняються досить часто, що може бути пояснене прагненням задовольняти стандартам, що розвиваються, W3C. Як і все кросмplatformні фреймворки не дуже підходить для створення додатків, що вимагають інтенсивних математичних обчислень або 3D Анімації.

6. Мобільні пристрої в курсах комп'ютерних наук

На цей момент для навчання розробці мобільних додатків створені навчальні програми в багатьох університетах і інститутах по усьому світу. Серед них можна назвати Smartphone Programming Course for Undergraduates "Mobile Application Development" в Wireless Institute, University of Notre Dame [8], Mobile Software Engineering Undergraduate Degree Programme, University of Glasgow [9], University of Illinois computer science project Mobile Learning Communities (MLC) [10].

6.1 Nordsecmob

Nordsecmob - магістерська програма по комп'ютерній безпеці й мобільним технологіям. Програма дає студентам широкі знання по різних аспектах сучасних архітектур обміну даними і їх розвитку. Ключовими областями навчального плану є мережні додатки й сервіси, інформаційна безпека й робота з мережею з використанням мобільних пристроїв. Магістри будуть здатні розробляти свої власні рішення для мобільного Internet.

Програма Nordsecmob є одним з магістерських курсів вищої європейської якості, обраних European Commission Erasmus Mundus Programme. Програми Erasmus Mundus призначені для студентів будь-якої країни миру. Учасниками програми вже стали багато студентів з Росії.

На сайті програми Nordsecmob [11] можна знайти повні тексти магістерських дисертацій її учасників. Частина магістерських досліджень виконана на мобільній платформі Android. Необхідно відзначити, що базовим університетом програми є Aalto University School of Science (раніше Helsinki University of Technology, TKK), Finland, а одним зі спонсорів програми є компанія Nokia. Із цими факторами зв'язана помітна орієнтованість програми на платформи Symbian і Meego (Maemo), що розвиваються компанією Nokia.

6.2 CMER

Centre for Mobile Education and Research (CMER) [12] створений на базі Department of Computing and Information Science, University of Guelph, Canada. Місією CMER є залучення на передовий рубіж прикладних досліджень у розробці новітніх додатків і сервісів для сприяння утвору в області мобільних пристроїв, а також інтеграція мобільних пристроїв у навчальний план комп'ютерних наук. CMER частково організований компанією RIM, виробником мобільної платформи Blackberry.

В CMER розроблений академічний цикл для інтеграції мобільних пристроїв у комп'ютерний утвір (CMER Academic Kit for Integrating Mobile Devices into Computing Education). На сайті CMER представлені докладні матеріали по академічному циклу. Опис циклу може бути знайдено в статті Mahmoud Q.H. і ін. "An Academic Kit for Integrating Mobile Devices into the CS Curriculum" [13].

Цикл включає навчальні матеріали для повного вступного курсу по розробці мобільних додатків, а також конкретні навчальні модулі для інтеграції мобільних пристроїв у курси програмної інженерії, розробки ігор, webсервісів, інформаційної безпеки й операційних систем. В описі циклу демонструються приклади його успішного впровадження в навчання в Department of Computing and Information Science, University of Guelph. В описі циклу відзначається, що у фактичному стандарті вищої комп'ютерної освіти ACM Computing Curricula [14] для США й Канади курс Wireless and Mobile Computing є факультативним у рамках галузі знань "Net-Centric Computing". Затверджується необхідність включення курсу по розробці мобільних додатків у сучасний обов'язковий навчальний план для комп'ютерних наук.

6.3 Відкриті платформи для мобільних пристроїв

У [15] розміщений ресурс, на якому концентруються матеріали російською мовою про застосування ОПП при розробці додатків для мобільних пристроїв і, насамперед, для, що інтенсивно розвиваються мобільних Інтернет пристроїв (Mobile Internet Devices - MID). Ресурс фокусується на Інтернет планшетах Nokia (NIT) на основі ОПП таємо і класичних платформах на базі ОС Symbian (S60). На ресурсі в першу чергу представляються навчальні матеріали (або посилання на них), призначені для освоєння методів розробки додатків студентами й початківцями роботи в середовищах відкритих програмних платформ (у тому числі й мобільних

7. Мобільні платформи як об'єкт наукових досліджень

Бурхливий індустріальний розвиток мобільних платформ у цей час має своєю основою економічний інтерес як великих компаній, так і окремих програмістів. Розробка додатків для мобільних платформ по більшій частині базується на накопиченому досвіді й "кращих практиках". Але систематичний науковий аналіз у цій новій сфері програмування проводився в набагато меншому ступені в порівнянні, приміром, з розробкою додатків для настільних комп'ютерів. У той же час при розробці мобільних додатків є цілий ряд особливостей, які немає необхідності враховувати при розробці інших типів додатків.

У своїй статті "Software Engineering Issues for Mobile Application Development" [17] Wasserman виділяє дві області досліджень в інженерії мобільного ПО: користувацький досвід (The User Experience) і нефункціональні вимоги (Non-functional Requirements). З користувацьким досвідом автор зв'язує наступні можливі напрямки досліджень:

- Як визначити, які функції повинні бути присутнім у мобільній версії традиційного додатка? Существуют ли техники, которые могут гарантировать максимальное переиспользование кода среди различных версий
- Які порівнянні зусилля для створення "native" мобільних додатків (або їх наборів для різних платформ) у порівнянні з мобільними webдодатками? чи є вимірювана відмінність у задоволенні користувачів або продуктивності між "native" і webдодатками?
- Як дизайнер ПО інтегрує різні форми введення й сенсори в дизайн додатка?

З нефункціональними вимогами автор зв'язує наступні можливі напрямки досліджень:

- Одинаково Одинаково ли ведут себя мобильные вебприложения в случае соединения с использованием телефонных сетей (3G, 4G), 802.11 (Wifi) і 802.16 (Wimax)? Есть Есть ли различия в безопасности Есть Есть ли значимые различия в устойчивости связи Адекватна Адекватна ли традиционная техника возврата сообщения об ошибках и управления исключениями или более высокая вероятность обрыва соединения (или прерывания связи) требует дополнительных механизмов обработки ошибочных ситуаций
- Нужны Нужны ли новые механизмы, чтобы гарантировать целостность данных, или достаточно традиционной техники синхронизации в клиентсерверных системах? Являются ли потенциальная возможность потери соединения или заряда батареи рисками для программы и/или целостности данных, если подобное событие произойдет в процессе транзакции или обновления системы
- Повинні чи додатки бути спроектовані по-різному залежно від швидкості передачі даних у мережі, у якій вони будуть використовуватися? В Азії деякі країни надають швидкість передачі даних 50 МБ/з або вище в той час, як типові швидкості, приміром, у США, навіть для мереж 3G, нижче 1 МБ. * Як розроблювач повинен створювати додаток, щоб максимально збільшити час роботи батареї й використання ресурсів?
- Це лише невеликий перелік можливих напрямків досліджень.

ВИСНОВОК

Оскільки практично всі сьогодні носять з собою смартфони і різні мобільні пристрої, мобільні програми дозволяють мати необхідну інформацію під рукою. Ці програми дозволяють ефективно інтегрувати інформацію з соціальними мережами, сайтом компанії, мультимедійним контентом і засобами комунікації. Будь-яка необхідна інформація може бути представлена в одному додатку і актуалізована для конкретної людини, місця і часу.

Такий спосіб концентрованого надання інформації на персональному мобільному пристрої дозволяє значно скоротити відстань між компанією, брендом і джерелом інформації з клієнтом, споживачем або співробітником. Це наділяє користувача додатка додатковими можливостями, які недоступні йому при умовах відсутності подібного програми. Компанія ж отримує можливість постійної комунікації з користувачем за допомогою одного каналу, одночасно включає в себе функції кількох.

Наявність власних брендovаних і корпоративних мобільних додатків, несучих собою різні функції, - це сьогодні просто необхідність для підвищення ефективності компанії в умовах сучасного ринку.

Крім того, зараз розробником програми може стати абсолютно кожен.

ЛІТЕРАТУРА:

ОСНОВНА

- [1] Mahmoud, Q.H., and Popowicz, P., "A Mobile Application Development Approach to Teaching Introductory Programming", 40th ASEE/IEEE Frontiers in Education Conference, Session T4F, October 27-30, Washington, DC.
- [2] Hartikainen, V., Liimatainen, P.P., and Mikkokek, T., "On Mobile Java Memory Consumption." Proceedings of the 14th Euromicro International Conference on Parallel, Distributed, and Network-based Processing (PDP2006), Montbeliard-Sochaux, France, pp. 333-339.
- [3] Kontio, M., "Better MIDlets by design." IBM developerWorks, www.ibm.com/developerworks/wireless/library/wi-design.html, просматривалось 24 января 2011.
- [4] iOS Dev Center:
<https://developer.apple.com/devcenter/ios/index.action#downloads>.
- [5] Download the Android SDK: <http://developer.android.com/sdk/index.html>.
- [6] Key Information for Developing Java Applications:
<http://us.blackberry.com/developers/javaappdev>.
- [7] Allen, S., and Graupera, V., and Lundrigan, L, "Pro Smartphone Cross-Platform Development".
- [8] A Smartphone Programming Course for Undergraduates:
<http://wireless.nd.edu/wi-nd-mobileapps-white-paper.pdf>.
- [9] Mobile Software Engineering Undergraduate Degree Programme:
<http://www.gla.ac.uk/undergraduate/degrees/mobilesoftwareengineering>.
- [10] University of Illinois computer science project Mobile Learning Communities (MLC): <http://cs.illinois.edu/news/2010/Jan12-1>.
- [11] NordSecMob: <http://nordsecmob.tkk.fi>.
- [12] CMER: <http://cmer.cis.uoguelph.ca>.
- [13] Mahmoud, Q.H., Ngo, T., Niazi, R., Popowicz, P., Sydoryshyn, R., Wilks, M., and Dietz, D., "An Academic Kit for Integrating Mobile Devices into the CS Curriculum.", Proceedings of the 14th ACM-SIGCSE Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2009), Paris, France, pp. 40-44. или <http://cmer.cis.uoguelph.ca/cmer-ak/AcademicKit-V1.0/publications/ITICSE2009-Paper.pdf>.
- [14] ACM Computing Curricula (ACM-CC 2001):
http://www.acm.org/education/education/education/curric_vols/cc2001.pdf, (обновлен в 2008).
- [15] Открытые платформы для мобильных устройств: <http://oss.fruct.org>.
- [16] ФГОС ВПО по специальности 090301 Компьютерная безопасность:

<http://www.edu.ru/mon-site/pro/fgos/vpo/pv090301s.pdf>.

[17] Anthony I. Wasserman, "Software Engineering Issues for Mobile Application Development", FoSER 2010, November 7-8, 2010, Santa Fe, New Mexico, USA, <http://www.cmu.edu/silicon-valley/wmse/wasserman-foser2010.pdf>.

ДОПОМІЖНА (ANDROID)

1. Розробка першого додатки для Windows Phone: Підготовка робочого оточення (EN). – (<http://msdn.microsoft.com/ru-ru/jj553273.aspx>)
2. Пакет Windows Phone SDK 7.1. (<http://www.microsoft.com/ru-ru/download/details.aspx?id=27570>)
3. Windows Phone 8 SDK Preview (EN). (<http://wp7forum.ru/windows-phone-8-sdk-preview-ne-dlya-vsex/>)
4. Get the Android SDK. (EN). (<http://developer.android.com/sdk/index.html?hl=sk>)
5. Android SDK for Real-Time Apps | PubNub (EN). (<http://www.pubnub.com/docs/java/android/android-sdk.html>)
6. Порівняльна характеристика Андроїд і iOS шляхом розгляду двох корпорацій-розробників. (<http://fans-android.com/sopostavitelnaya-harakteristika-android-i-ios-putem-sravneniya-dvuh-korporatsiy-razrabotchikov-google-i-apple-planyi-google-po-peremanivaniyu-yablochnikov/>)
7. Android Media Playback (EN). (<http://developer.android.com/guide/topics/media/mediaplayer.html>)
8. Apple - iOS 7 - Что такое iOS (<https://www.apple.com/ru/ios/what-is/>)
9. Перспективи розробки під Android (EN). (http://armikael.com/mobile/prospects_of_development_on_android.html)
10. Розробка під iOS: 60% (EN). (<http://habrahabr.ru/company/apps4all/blog/143470/>)