**МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ**

**Харківський національний університет внутрішніх справ**

**Факультет № 4**

**Кафедра інформаційних технологій**

# ТЕКСТ ЛЕКЦІЇ

**з дисципліни «Розробка захищених мобільних застосувань»**
**за темою «Розробка мобільних додатків на платформі Android»**

**Галузь знань: 12 "Інформаційні технології "**

**Спеціальність: 125 "Кібербезпека"**

**Ступінь вищої освіти - магістр**

м. Харків
2017 р.

## Передмова

**СХВАЛЕНО**
Науково-методичною радою ХНУВС
_____ Протокол № _____
(дата, місяць, рік )

**ЗАТВЕРДЖЕНО**
Вченою радою факультету № 4 ХНУВС
_____ Протокол № _____
(дата, місяць, рік )

_____ _____
(підпис) (П.І.Б.)

**ПОГОДЖЕНО**
Секцією Науково-методичної ради ХНУВС з технічних дисциплін
_____ Протокол № _____
(дата, місяць, рік )

_____ _____
(підпис) (П.І.Б.)

**ЗАТВЕРДЖЕНО**
На засіданні кафедри інформаційних технологій
_____ Протокол № _____
(дата, місяць, рік )

_____ _____
(підпис) (П.І.Б.)

**Рецензент:**
**Зацеркляний М.М.,** доктор технічних наук, професор;

Розробники: Янголенко О.В. – Харків: Харківський національний університет внутрішніх справ, 2017

## План лекції

1. Підходи до розробки мобільного додатку на платформі Android.
2. Інструментарій та бібліотеки розробки мобільних додатків на платформі Android.
3. Бази даних для Android-додатка.

## Література:

Основна:

1. Дейтел П. Android для программистов: Создаём приложения / П. Дейтел, Х Дейтел, Э. Дейтел, М. Моргано. — СПб.: Питер, 2013. – 560 с.

2. Голощапов А. Л. Google Android: программирование для мо-бильных устройств / Голощапов А. Л. – СПб. : БХВ-Петербург, 2011. – 448 с. : ил. + CD-ROM – (Профессиональное программирование).

3. С. Хашими, С. Разработка приложений для Android / С. Хашими, С. Коматинени, Д. Маклин – СПб. : Питер, 2011. – 736 с. : ил.

4. Kurniawan B. Android application development: A beginner's tutorial / B. Kurniawan. – Brainy Software, 2015.

Додаткова:

5. Майер Р. Android 2: программирование для планшетных компьютеров и смартфонов / Ретро Майер ; пер. с англ.– М. : Эксмо, 2011. – 672 с.

6. Дэрси Л. Android за 24 часа. Программирование приложений под операционную систему Google / Дэрси Л., Кондер Ш. – М. : Рид Групп, 2011. – 464 с.

## Текст лекції

### *1. Підходи до розробки мобільного додатку на платформі Android.*

Існує два основні підходи для розробки мобільних додатків: розробка нативних додатків з використання мови програмування для розроблюваної платформи і створення гібридного додатка з використанням CSS, HTML і JavaScript.

Нативні додатки – це програми, розроблені під одну з мобільних платформ: Apple iOS, Google Android, Windows Phone і т.д. Основна перевага нативних додатків полягає в тому, що вони оптимізують призначений для користувача досвід, додаток працює максимально швидко, тому що він був розроблений для конкретної платформи. У той же час прив'язка до платформи є і головним її недоліком – якщо додаток необхідно розробити для більш ніж однієї платформи, його перенесення на кожну наступну мобільну операційну систему починається практично з нуля.

Гібридними називаються додатки, які розробляються за допомогою HTML, CS і JavaScript – тобто тими ж технологіями, що й Інтернет сайти, а

для того щоб вони могли бути встановлені на смартфоні, код «обертається» в нативну оболонку. Для цієї процедури існує кілька як платних, так і безкоштовних рішень, найпоширенішими є PhoneGap і Cordova. При виборі технології створення мобільних додатків перевага віддається нативним додаткам, особливо якщо потрібна велика швидкість відгуку, обробка та відображення великих обсягів інформації, доступ до апаратного забезпечення.

Додатки для Android складаються з компонентів, котрі система може запускати і керувати так, як їй необхідно. Головна особливість платформи Android полягає в тому, що один додаток може використовувати елементи інших додатків (за умови, що ці програми дозволяють їх використовувати). При цьому ваш додаток не включає код іншого додатка або посилання на нього, а просто запускає потрібний елемент іншої програми. Для реалізації такого використання компонентів інших додатків система повинна бути в змозі запустити процес для додатка, в якому знаходиться необхідний компонент, і форматувати потрібні їй об'єкти. Тому, на відміну від додатків в більшості інших систем, у додатків Android немає єдиної точки входу для запуску всієї програми, аналогічної, наприклад, функції main () в подібних мовах програмування. Androidдодатки складаються з компонентів, які система може ініціювати і запустити при необхідності.

Додатки на Android складаються з таких компонентів:

- Activity є зовнішнім інтерфейсом для однієї операції, що може зробити користувач. Являє собою схему представлення Androidдодатків. Наприклад, екран, який бачить користувач. Androidдодаток може мати кілька активів і може перемикатися між ними під час виконання програми;
- Views. Користувацький інтерфейс активіти, створений віджетами класів, успадкованих від «android.view.View». Схема views керується через «android.view.ViewGroups»;
- Services – являє собою компоненти, що працюють в фоновому режимі. Виконує фонові завдання без надання користувацького інтерфейсу. Вони можуть повідомляти користувача через систему повідомлень Android. Service, в основному, потрібний при довгих операціях або для забезпечення роботи віддалених процесів;
- Content Provider. Надає дані додатків, за допомогою контент-провайдера. Ваша програма може обмінюватися даними з іншими додатками. Android містить базу даних SQLite, яка може бути контент-провайдером;
- Intents. Асинхронні повідомлення, які дозволяють додатку запросити функції з інших служб або активи. Додаток може робити прямі Intents службі або активи чи запитати у Android зареєстровані служби і додатки для Intents. Наприклад, додаток може запитати через Intents контакт з програми контактів (телефонної записної книжки) апарату. Додаток реєструє саме себе в Intents через IntentFilter. Intents – потужний концепт, що дозволяє створювати слабкозв'язані додатки;
- Broadcast Receiver. Широкомовний приймач – приймає системні

повідомлення і неявні інтенти, може використовуватися для реагування на зміну стану системи. Додаток може реєструватися як приймач певних подій і може бути запущений, якщо така подія відбулася. Іншими частинами Android являються віджети, живі папки (Live Folders), або живі шпалери (Live Wallpapers). Живі папки відображають джерело будь-яких даних на «робочому столі» без запуску відповідних додатків. Таким чином, проаналізувавши компоненти додатків ОС Android та дослідивши технології їх створення під цю операційну систему, ми можемо зробити висновок, що ОС Android – найпопулярніша операційна система на сьогоднішній день. І для того, щоб вона повноцінно працювала, розробляється ряд різних додатків. Саме тому потрібно знати технології створення мобільних додатків та компоненти з яких вони складаються. У подальшому планується розробити мобільний додаток редагування аудіозаписів на Android.

Розглянемо детальніше компонент активності. Активність може бути повернена у працюючий стан з режимів паузи або зупинки. Активність в будь-якому стані є одним і тим же Java-екземпляром у пам’яті, тому її вміст залишається в тому ж стані, що і перед зупинкою або припиненням

У активності є кілька захищених методів, які можна перевизначити для отримання інформації про зміни стану

ActivityOnCreate – метод викликається при першому запуску активності У ньому ми встановлюємо компоненти інтерфейсу і звертаємося до системи вводу Цей метод викликається всього один раз протягом життєвого циклу активності.

ActivityOnRestart – метод викликається при поверненні активності із зупиненого стану. Йому має передувати виклик методу onStop.

ActivityOnStartC – метод викликається після методу onCreate або коли активність повертається з зупиненого стану. У другому випадку йому передує виклик методу onRestart.

ActivityOnResume – метод викликається після виведення методу onStartO або коли активність повертається з призупиненого стану (наприклад, при розблокуванні екрану).

Activity onPauseO – метод викликається при переході активності в пріостанов-х ленное стан Це може виявитися останнім отриманим нами оповіщенням, оскільки ОС може вирішити закрити наш додаток без повідомлення. Тому в цьому методі нам необхідно зберегти всі важливі для нас параметри програми.

ActivityOnStop – метод викликається при переході активності в режим зупинки Йому передує виклик методу onPause – перед переходом активності в зупинене стан вона завжди спочатку проходить режим паузи. Як і у випадку з onPause, це може бути останній момент, коли ми отримуємо інформацію від програми перед його закриттям операційною системою. Тут ми також можемо зберегти поточний стан. Однак система може вирішити не викликати цей метод і просто знищити активність Тому, оскільки перед onStop і мовчазним закриттям програми завжди викликається onPause, збереження важливої для нас інформації найкраще доручити методу onPause.

ActivityOnDestroy – метод викликається в кінці життєвого циклу активності, коли вона остаточно знищується. Це остання точка, в якій ми можемо отримати і зберегти якісь дані, які хочемо отримати назад при пересоздании нашої активності Зверніть увагу – цей метод може ніколи не викликатися, якщо активність була знищена системою в неявному режимі після виклику onPauseC або onStopO

## *2. Інструментарій та бібліотеки розробки мобільних додатків на платформі Android.*

Програми для Android є програмами в нестандартному байт-коді для віртуальної машини Dalvik. Dalvik оптимізований для низького споживання пам'яті, це нестандартна регістр-орієнтована віртуальна машина, яка добре підходить для виконання на RISC-архітектурах процесорів, котрі часто використовуються у мобільних та вбудованих пристроях, таких, як комунікатори й планшетні комп'ютери. Більшість віртуальних машин, що використовуються на десктопах, є стек-орієнтованими, включаючи стандартну віртуальну машину Java від Sun/Oracle.

Програми для Dalvik пишуться на мові Java. Попри це, стандартний байт-код Java не використовується, замість нього Dalvik VM виконує байт-код власного формату. Після компіляції сирцевих текстів програми на Java (за допомогою javac) утиліта dx з «Android SDK» перетворює .class файли у формат .dex, придатний для інтерпретації в Dalvik.

З бібліотек класів Dalvik не застосовує ані Java SE, ані Java ME Class Library (в тому числі, класи Java ME, AWT та Swing не підтримуються). Замість цього використовується своя власна бібліотека, побудована на підмножині Java-реалізації Apache Harmony.

Google пропонує для вільного завантаження інструментарій для розробки (Software Development Kit), який призначений для x86-машин під операційними системами Linux, Mac OS X (10.4.8 або вище), Windows XP, Windows Vista та Windows 7. Для розробки потрібен Java Development Kit 5 або новіший.

Розробку застосунків для Android можна вести мовою Java (не нижче Java 1.5). Офіційним середовищем розробки є Android Studio, представлене компанією Google в 2013. Крім цього існує плагін для Eclipse — «Android Development Tools» (ADT), призначений для Eclipse версій 3.3-3.7. Для IntelliJ IDEA також існує плагін, який полегшує розробку Android-застосунків. Для середовища розробки NetBeans розроблено плагін, який починаючи з версії Netbeans 7.0 перестав бути експериментальним. Крім того існує Motodev Studio for Android, що являє собою комплексне середовище розробки, засноване на базі Eclipse і дозволяє працювати безпосередньо з Google SDK.

Крім того в 2009 році на застосунок до ADT був опублікований Android Native Development Kit (NDK), пакет інструментаріїв і бібліотек дозволяє вести розробку застосунків мовою C/C++. NDK рекомендується використовувати для розробки ділянок коду, критичних до швидкості.

Доступні бібліотеки:

- Bionic (Бібліотека стандартних функцій, несумісна з libc);
- libc (стандартна системна бібліотека мови Сі);
- мультимедійні бібліотеки (на базі PacketVideo OpenCORE; підтримують такі формати, як MPEG-4, H.264, MP3, AAC, AMR, JPEG та PNG);
- SGL (рушій двовимірної графіки);
- OpenGL ES 1.0 (рушій тривимірної графіки);
- Surface Manager (забезпечує для застосунків доступ до 2D/3D);
- WebKit (готовий рушій для Web-браузера; обробляє HTML, JavaScript);
- FreeType (рушій обробки шрифтів);
- SQLite (проста система керування базами даних, доступна для всіх застосувань);
- SSL (протокол, що забезпечує безпечну передачу даних по мережі).

В порівнянні із звичайними застосунками Linux, застосунки Android vмають додаткові особливості:

- Content Providers — обмін даними між застосунками;
- Resource Manager — доступ до таких ресурсів, як файли XML, PNG, JPEG;
- Notification Manager — доступ до рядка стану;
- Activity Manager — управління активними застосунками;
- Для Android був розроблений формат інсталяційних пакетів .apk.

### *3. Бази даних для Android-додатка.*

Мобільні додатки Android найчастіше використовують в якості бази даних SQLite. Особливістю SQLite є те, що вона не використовує парадигму клієнт-сервер, тобто рушій SQLite не є окремим процесом, з яким взаємодіє застосунок, а надає бібліотеку, з якою програма компілюється і рушій стає складовою частиною програми. Таким чином, як протокол обміну використовуються виклики функцій (API) бібліотеки SQLite. Такий підхід зменшує накладні витрати, час відгуку і спрощує програму. SQLite зберігає всю базу даних (включаючи визначення, таблиці, індекси і дані) в єдиному стандартному файлі на тому комп'ютері, на якому виконується застосунок. Простота реалізації досягається за рахунок того, що перед початком виконання транзакції весь файл, що зберігає базу даних, блокується; ACID-функції досягаються зокрема за рахунок створення файлу-журналу.

Кілька процесів або потоків можуть одночасно без жодних проблем читати дані з однієї бази. Запис в базу можна здійснити тільки в тому випадку, коли жодних інших запитів у цей час не обслуговується; інакше спроба запису закінчується невдачею, і в програму повертається код помилки. Іншим варіантом розвитку подій є автоматичне повторення спроб запису протягом заданого інтервалу часу.

У комплекті постачання йде також функціональна клієнтська частина у вигляді виконуваного файлу sqlite3, за допомогою якого демонструється реалізація функцій основної бібліотеки. Клієнтська частина працює

з командного рядка, і дозволяє звертатися до файлу БД на основі типових функцій ОС.

Завдяки архітектурі рушія можливо використовувати SQlite як на вбудовуваних (embedded) системах, так і на виділених машинах з гігабайтними масивами даних.

Особливості SQLite:

- Транзакції атомарні, послідовні, ізольовані, і міцні (ACID) навіть після збоїв системи і збоїв живлення.
- Встановлення без конфігурації — не потребує ані установки, ані адміністрування.
- Реалізує значну частину стандарту SQL92.
- База даних зберігається в одному крос-платформовому файлі на диску.
- Підтримка терабайтних розмірів баз даних і гігабайтного розміру рядків і BLOBів.
- Малий розмір коду: менше ніж 350KB повністю налаштований, і менш 200KB з опущеними додатковими функціями.
- Швидший за популярні рушії клієнт-серверних баз даних для найпоширеніших операцій.
- Простий, легкий у використанні API.
- Написана в ANSI C, включена прив'язка до TCL; доступні також прив'язки для десятків інших мов.
- Добре прокоментований сирцевий код зі 100% тестовий покриттям гілок.
- Доступний як єдиний файл сирцевого коду на ANSI C, який можна легко вставити в інший проект.
- Автономність: немає зовнішніх залежностей.
- Крос-платформовість: з коробки підтримується Unix (Linux і Mac OS X), OS/2, Windows (Win32 і WinCE). Легко переноситься на інші системи.
- Поставляється з автономним клієнтом інтерфейсу командного рядка, який може бути використаний для управління базами даних SQLite.

Сама бібліотека SQLite написана мовою C; існує велика кількість прив'язок до інших мов програмування, зокрема до C++, Java, Python, Perl, PHP, Tcl (засоби для роботи з Tcl включені в комплект постачання SQLite), Ruby, Haskell, Scheme, Smalltalk і Lua, а також до багатьох інших.