

УДК 004.4:621.395:336

Єрмейчук С. Ю., Федонюк А.А. к.ф-мат.наук, доцент
СНУ імені Лесі Українки

РОЗРОБКА АГРЕГАТОРА КУРСА ВАЛЮТ ДЛЯ ОС ANDROID

Єрмейчук С. Ю., Федонюк А.А. Розробка агрегатора курса валют для ОС Android. Програмування для мобільних пристроїв – важлива галузь інформаційних технологій. Для цього сегмента прогнозують швидкий ріст у міру того, наскільки мобільні програми будуть підтримані технічними можливостями мобільних пристроїв. Інформаційні технології (ІТ) зазнали глобального поширення у всіх галузях: від освіти і до медицини. Однією з найбільш популярних та таких, що швидко зростає в ІТ є галузь *wearable technology*, що в перекладі на українську мову – носимі технології. Зазвичай це предмети одягу або аксесуари (все те, що людина може носити на собі), що є одночасно і електронними пристроями. Зазвичай такого роду пристрої називають носимими комп'ютерами (*wearable computers, wearables*).

Ключові слова: **Android; віджети; мобільний додаток; Android Studio; агрегація; бібліотека Volley; Android SDK.**

Єрмейчук С. Ю., Федонюк А.А. Разработка агрегатора курса валют для ОС Android. Программирование для мобильных устройств - важная отрасль информационных технологий. Для этого сегмента прогнозируют быстрый рост по мере того, как мобильные приложения будут поддержаны техническими возможностями мобильных устройств. Информационные технологии (ИТ) получили глобального распространения во всех областях: от образования и к медицине. Одной из наиболее популярных и таких быстро растущих в ИТ является отрасль *wearable technology*, что в переводе - носимые технологии. Обычно это предметы одежды или аксессуары (все то, что человек может носить на себе), что является одновременно и электронными устройствами. Обычно такого рода устройства называют носимый компьютер (*wearable computers, wearables*).

Ключевые слова: **Android; виджеты; мобильное приложение; Android Studio; агрегация; библиотека Volley; Android SDK.**

Yermeychuk S.Y, Fedonyuk A.A Development aggregator exchange rates for Android OS. Programming for mobile devices – is an important area of informational technology. This segment predicts rapid growth as how mobile applications will be supported by the technical capabilities of mobile devices. Informational technology (IT) has global distribution in all areas, from education to medicine. One of the most popular and the fastest growing in the IT industry is a *wearable technology*, which is translated into Ukrainian as the technology you can take with yourself. Usually it is a part of clothes or accessories (everything that people can carry with themselves), which is an electronic device at the same time. Usually this type of device is called as a *wearable computer (wearable computers, wearables)*.

Keywords: **Android; widgets; mobile app; Android Studio; aggregation; Library Volley; Android SDK.**

Постановка наукової проблеми та її значення. У сучасному світі мобільні електронні пристрої поширюються дуже швидко. Потужності збільшуються в геометричній прогресії. Ринок техніки все більше тяжіє до портативних пристроїв, тому програмісти розробляють для них велику кількість професійних додатків. Раніше мобільні додатки використовувалися для швидкої перевірки електронної пошти, але їх високий попит призвів до розширення їхніх призначень і в інших областях, таких як ігри для мобільних телефонів, GPS, спілкування, перегляд відео та користування інтернетом.

Мобільний додаток – це програмне забезпечення, призначене для роботи на смартфонах, планшетах та інших мобільних пристроях. Багато мобільних додатків вже встановлені на самому пристрої або можуть бути завантажені на нього з додатків онлайн-магазинів, таких як App Store, BlackBerry App World та інших. Тому створення мобільних додатків на ОС Android - популярний напрям сучасних ІТ-технологій. ОС Android встановлена на мільйонах девайсів: смартфонах, комунікаторах, планшетах, нетбуках та ін. Її особливість в тому, що це відкрита та безкоштовна платформа, на яку кожен може встановити різноманітні ігри, додатки для бізнесу, спілкування та відеозв'язку, медіа ресурсів та ЗМІ, онлайн-ресурсів.

Метою нашого дослідження є створення Android-додатку для перегляду курсів валют через встановлене з'єднання з мережею Internet У відповідності до поставленої мети визначено такі завдання дослідження: ознайомитись з операційною системою Android та її специфікаціями; розглянути процес моделювання роботи Android-додатку перегляду курсів валют; організувати збір статистичних даних з банків; розглянути механізм передачі даних через мережу Інтернет; розглянути і дослідити роботу стандарту передачі даних JSON; розробити програмний додаток на базі ОС Android; забезпечити зручне подання даних.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження. Android – операційна система від Google для мобільних пристроїв, таких як планшетні

комп'ютери, комунікатори, мобільні телефони, камери та багато інших. Вона створена на базі ядра Linux. Назва «Android» в перекладі з англійської означає «людиноподібний робот». Спочатку операційна система розроблялася компанією Android Inc., яку згодом купила Google. Потім Google створила Open Handset Alliance - організацію, яка займається підтримкою і розвитком платформи. Розробка додатків для Android здійснюється, як правило, на мові Java, але можна використовувати і інші мови програмування. З моменту виходу першої версії у вересні 2008 року відбулося 40 оновлень системи. Ці оновлення, як правило, стосуються виправлення виявлених помилок і додавання нової функціональності в систему. Остання версія, на момент написання роботи — 6.0. В таблиці 1 наведені дати виходу основних версій Android.

Платформа Android представляє собою програмний стек для мобільних пристроїв, який включає операційну систему, програмне забезпечення проміжного шару (middleware), а також основні користувальницькі додатки, що входять до складу мобільного телефону, календар, карти, браузер, бази даних контактів, повідомлень SMS та ін.

Архітектура Android складається з чотирьох рівнів:

1. рівень ядра;
2. рівень бібліотек і середовища виконання;
3. рівень каркаса додатків (application framework);
4. рівень додатків.

Таблиця 1 - Список версій OS Android

Версія	Версія ядра Linux	Дата виходу	Кодова назва
1.0	2.6.25	23 вересня 2008 року	Apple Pie(API 1)
1.1	2.6.25	09 лютого 2009 року	Banana Bread (API 2)
1.5	2.6.27	30 квітня 2009 року	Cake(API 3)
1.6	2.6.29	15 вересня 2009 року	Donut(API 4)
2.0.x/2.1	2.6.29	27 жовтня 2009 року 12 січня 2010 року	Eclair(API 5,6,7)
2.2.x	2.6.32	20 травня 2010 року	Froyo(API 8)
2.3.x	2.6.35	06 грудня 2010 року	Gingerbread(API 9,10)
3.0	2.6.36	03 лютого 2011 року	Honeycomb(API 11,12,13)
4.0	3.0	19 жовтня 2011 року	Ice Cream Sandwich (API 14,15)
4.1 4.2 4.3	3.0.8-3.4 (4.2.1+)	27 червня 2012 року 29 жовтня 2012 року 24 липня 2013 року	Jelly Bean(API 16,17,18)
4.4	3.4	31 жовтня 2013 року	KitKat(API 19)
5.0	3.4	03 листопада 2014 року	Lollipop(API 21)
5.1	3.4	9 березня 2015 року	LolliPop(API 22)
6.0	3.10	28 травня 2015 року	Marshmallow(API 23)

На рисунку 1. показані основні компоненти операційної системи Android і їх взаємодія.

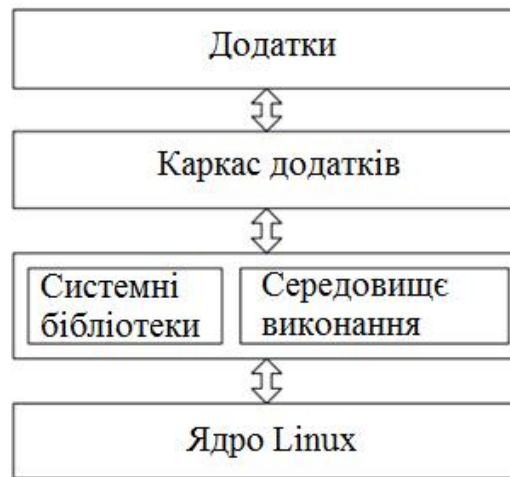


Рисунок 1 - Архітектура системи Android

На рівні ядра розташовуються основні служби типу управління процесами, розподілу пам'яті і управління файловою системою. Android заснований на ядрі Linux версії 2.6, але сама система Android не є Linux-системою в «чистому вигляді». Система Android має деякі відмінності і містить додаткові розширення ядра Linux, специфічні для Android, - свої механізми розподілу пам'яті, взаємодія між процесами і ін.

Наступний рівень над ядром Linux включає набір бібліотек C/C++, використовуваних різними компонентами ОС. Бібліотеки цього рівня за своїм функціональним призначенням можна розділити на дві групи:

- системну бібліотеку C;
- функціональні бібліотеки C/C ++.

Системна бібліотека базується на Berkeley Software Distribution (BSD). Компанія Google розробила власну версію системної бібліотеки libc – Bionic спеціально для мобільних пристроїв на основі Linux. Це було необхідно для забезпечення швидкого завантаження бібліотеки в кожен процес, і отже, бібліотека повинна була мати маленький розмір. Бібліотека Bionic має розмір близько 200 Кбайт, що в два рази менше розміру стандартної бібліотеки Linux glibc. Даний рівень представлений бібліотеками libc (в Android вона називається Bionic), OpenGL (підтримка графіки), WebKit (відображення Web-сторінок), FreeType (підтримка шрифтів), SSL (зашифровані з'єднання), SGL (2D-графіка), бібліотеки підтримки SQLite, Media Framework (підтримка мультимедіа). На цьому ж рівні працює Dalvik Virtual Machine - віртуальна машина Java, яка надає необхідну функціональність для Java-додатків.

Рівень каркаса додатків знаходиться на вершині системних бібліотек, функціональних бібліотек і Dalvik VM. На цьому рівні знаходяться основні служби Android для управління життєвим циклом додатків, пакетами, ресурсами і т.д.

Мобільний пристрій Android поставляється з набором основних додатків, включаючи поштового клієнта, програму для роботи з SMS, календар, навігаційні карти, браузер, контакти та ін [1].

Неможливо створити осмислений додаток, не вивчивши внутрішню організацію, властиву додаткам, які працюють на певній платформі. В першу чергу, необхідно проаналізувати структуру та основні компоненти додатків, що розробляються для роботи на пристроях під управлінням ОС Android. Від типу мобільного пристрою внутрішня організація додатків не залежить, тому Android-додатки, що розроблені для смартфонів, цілком зможуть виконуватися і на планшетах. При проектуванні та розробці програмного забезпечення має велике значення його швидкість та ефективність роботи. Дуже важливо постійно приділяти увагу оптимізації програмного забезпечення на всіх етапах його розробки. Для цього потрібно вимірювати час виконання елементів програмного коду, аналізувати хід роботи програмного забезпечення, виділяти обмежені ділянки програмного коду, слідкувати за виділенням пам'яті та часом життя об'єктів.

Приступаючи до розробки мобільних додатків, якщо вдасться визначити до якого типу належить додаток, то стане зрозуміло на які моменти в процесі його розробки необхідно звертати основну увагу. Наступні види додатків:

1) Змішані додатки більшу частину часу працюють у фоновому режимі, проте допускають взаємодію з користувачем після. Зазвичай взаємодія з користувачем зводиться до повідомлення про які-небудь події. Прикладами таких додатків можуть служити мультимедіа-програвачі, програми для обміну текстовими повідомленнями (чати), поштові клієнти. Можливість реагувати на користувача введення і при цьому не втрачати працездатності у фоновому режимі є характерною особливістю змішаних додатків. Такі програми зазвичай містять як видимі активності, так і приховані (фонові) сервіси, і при взаємодії з користувачем повинні враховувати свій поточний стан.

2) Фонові програми після настройки не припускають взаємодії з користувачем, більшу частину часу перебувають і працюють в прихованому стані. Прикладами таких додатків можуть служити, служби екранування дзвінків, SMS-автовідповідачі. У більшості фонові програми націлені на відстеження подій, породжуваних апаратним забезпеченням, системою або іншими додатками, працюють непомітно.

3) Додатки переднього плану виконують свої функції тільки, коли видимі на екрані, в іншому ж випадку їх виконання призупиняється. Такими додатками є, наприклад, ігри, текстові редактори, відеопрогравачі.

4) Віджети - невеликі додатки, які відображаються у вигляді графічного об'єкта на робочому столі. Прикладами можуть служити, додатки для відображення динамічної інформації, такої як заряд батареї, прогноз погоди, дата і час.

Як вже було зазначено, програми під Android розробляються на мові програмування Java, компілюються у файл з розширенням .apk. Після цього цей файл використовується для установок програм на пристрої, що працюють під управлінням Android. Після установки кожен Android-додаток функціонує у своїй власній безпечній області. Розглянемо, як це виглядає:

- операційна система Android є багатокористувацькою ОС, в якій кожен додаток розглядається як окремий користувач;

- за замовчуванням, система призначає кожному з додатків унікальний користувальницький ID, який використовується тільки системою і невідомий самому додатку;

- система встановлює права доступу до всіх файлів додатка наступним чином: доступ до елементів додатку має тільки користувач з відповідним ID;

- кожному додатку відповідає окремий Linux процес, який запускається, як тільки це необхідно хоча б одному компоненту додатку. Процес припиняє роботу, коли жоден компонент програми не використовує його або ж системі потрібно звільнити пам'ять для інших (можливо, більш важливих) додатків;

- кожному процесу відповідає окремий екземпляр віртуальної машини Dalvik. У зв'язку з цим код програми виконується ізольовано від інших додатків.

Перераховані ідеї функціонування програм в ОС Android реалізують «принцип мінімальних привілеїв». Кожному додатку за замовчуванням, дозволений доступ тільки до компонентів, необхідних для його роботи і ніяким більше. Таким чином забезпечується дуже безпечно середовище функціонування додатків. Однак, у випадку необхідності додатки можуть отримати доступ до даних інших додатків і системних сервісів (послуг). У випадку, коли двом додаткам необхідно мати доступ до файлів один одного, їм присвоюється один і той же користувальницький ID. Для економії системних ресурсів такі додатки запускаються в одному Linux процесі і ділять між собою один і той же екземпляр віртуальної машини. В цьому випадку додатки також повинні бути підписані одним сертифікатом. У разі ж, коли додатку потрібно доступ до системних даних, наприклад, контактів, SMS повідомленнями, картам пам'яті, камері, Bluetooth і т.д., користувачеві необхідно дати додатком такі повноваження під час встановлення його на пристрій [2].

Програмний інструментарій Java на базі платформи Android. Для даної розробки вирішено зупинитися на мові Java, як офіційній та найбільш поширеній мові для Android. Для створення додатків за допомогою Java існує докладна документація на офіційному сайті Android розробки, також більшість навчальних матеріалів створено з розрахунком саме на неї.

Java — об'єктно-орієнтована мова програмування, випущена компанією Sun Microsystems у 1995 році як основний компонент платформи Java. Зараз мовою займається компанія Oracle, яка придбала Sun Microsystems у 2009 році. Програми на Java транслюються в байт-код, що виконується віртуальною машиною Java, яка обробляє байтовий код і передає інструкції обладнанню як інтерпретатор. Перевага подібного способу виконання програм полягає в повній

незалежності байт-коду від операційної системи і устаткування, що дозволяє виконувати Java-додатки на будь-якому пристрої, для якого існує відповідна віртуальна машина. Іншою важливою особливістю Java є система безпеки завдяки тому, що виконання програми повністю контролюється віртуальною машиною. Будь-які операції, які перевищують встановлені повноваження програми (наприклад, спроба несанкціонованого доступу до даних або з'єднання з іншим комп'ютером) викликають негайне переривання. Слід зауважити, що фактично, більшість архітектурних рішень, прийнятих при створенні Java, було продиктовано бажанням надати синтаксис, схожий з C/C++.

Огляд існуючих середовищ програмування Java. В якості інтегрованого середовища розробки в основному виступає Android Studio. Воно було представлено 16 травня 2013 року на конференції Google I/O, довго перебувало на стадії бета-тесту, а перша стабільна версія вийшла 8 грудня 2014 року. На цей час для завантаження доступні версії Android Studio 1.2 та Android Studio 1.3 Preview [6].

Android Studio прийшов на зміну плагіну ADT для платформи Eclipse. Середовище побудоване на базі продукту IntelliJ IDEA Community Edition, що розробляється компанією JetBrains. Android Studio розвивається в рамках відкритої моделі розробки та поширюється під ліцензією Apache 2.0. Бінарні складання підготовлені для Linux (для тестування використаний Ubuntu), Mac OS X і Windows. Середовище надає засоби для розробки програм не тільки для смартфонів і планшетів, але і для носимих пристроїв на базі Android Wear, телевізорів (Android TV), окулярів Google Glass і автомобільних інформаційно-розважальних систем (Android Auto). Для додатків, спочатку розроблених з використанням Eclipse і ADT Plugin, підготовлений інструмент для автоматичного імпорту існуючого проекту в Android Studio.

Середовище розробки адаптоване для виконання типових завдань, що вирішуються в процесі розробки додатків для платформи Android. У тому числі у середовище включені засоби для спрощення тестування програм на сумісність з різними версіями платформи та інструменти для проектування додатків, що працюють на пристроях з екранами різної роздільності (планшети, смартфони, ноутбуки, годинники, окуляри тощо). Крім можливостей, присутніх в IntelliJ IDEA, в Android Studio реалізовано кілька додаткових функцій, таких як нова уніфікована підсистема складання, тестування і розгортання додатків, заснована на складальному інструментарії Gradle і підтримуюча використання засобів безперервної інтеграції.

Для прискорення розробки додатків представлена колекція типових елементів інтерфейсу і візуальний редактор для їхнього компонування, що надає зручний попередній перегляд різних станів інтерфейсу програми (наприклад, можна подивитися як інтерфейс буде виглядати для різних версій Android і для різних розмірів екрану). Для створення нестандартних інтерфейсів присутній майстер створення власних елементів оформлення, що підтримує використання шаблонів. У середовище вбудовані функції завантаження типових прикладів коду з GitHub. До складу також включені пристосовані під особливості платформи Android розширені інструменти рефакторингу, перевірки сумісності з минулими випусками, виявлення проблем з продуктивністю, моніторингу споживання пам'яті та оцінки зручності використання. У редактор доданий режим швидкого внесення правок. Система підсвічування, статичного аналізу та виявлення помилок розширена підтримкою Android API. Інтегрована підтримка оптимізатора коду ProGuard. Вбудовані засоби генерації цифрових підписів. Надано інтерфейс для управління перекладами на інші мови.

Також можна використовувати Eclipse, IntelliJ IDEA чи NetBeans з відповідними плагінами та за наявності встановленого Android SDK, наприклад для середовища Eclipse випускається плагін Android Development Tools (ADT), що дозволяє створювати додатки та тестувати їх у вбудованому емуляторі операційної системи.

На сьогодні найбільш зручним та логічним рішенням є використання саме Android Studio. Це єдине середовище, що розроблялось спеціально для створення додатків для платформи Android. Воно включає в себе весь необхідний інструментарій для розробки, включаючи емулятор Android Virtual Device та віртуалізатор Intel Hardware Accelerated Execution Manager (HAXM) для прискорення роботи емулятора не потребує жодних плагінів чи налаштувань для роботи та використовує Gradle. До того ж це повністю офіційне середовище, яке розробляється і підтримується компанією Google. Android Studio дозволяє імпорт проектів, які розроблялися за допомогою IDE Eclipse з плагіном ADT [3].

В об'єктно-орієнтованому програмуванні під агрегуванням розуміють методику створення нового класу з уже існуючих класів шляхом включення.

Вкладені об'єкти нового класу зазвичай оголошуються закритими, що робить їх недоступними для прикладних програмістів, які працюють з класом. З іншого боку, творець класу може змінювати ці об'єкти, не порушуючи при цьому роботи існуючого клієнтського коду. Крім того, заміна вкладених об'єктів на стадії виконання програми дозволяє динамічно змінювати її поведінку. Механізм успадкування такою гнучкістю не володіє, оскільки для похідних класів встановлюються обмеження, що перевіряються на стадії компіляції.

На базі агрегування реалізується методика делегування, коли поставлено перед зовнішнім об'єктом завдання, яке передається внутрішньому об'єкту, що спеціалізується на вирішенні завдань такого роду.

Агрегація (агрегування за посиланням) – відношення «частина-ціле» між двома рівноправними об'єктами, коли один об'єкт (контейнер) має посилання на інший об'єкт. Обидва об'єкти можуть існувати незалежно: якщо контейнер буде знищений, то його вміст – ні.

Android система реалізує принципи найменших привілеїв. Тобто, кожен додаток, за замовчуванням, має доступ тільки до компонентів, які він вимагає, але не більше того. Це створює дуже безпечне середовище, в якій додаток не може отримати доступ до частини системи, до якої не дозволено торкатись.

Тим не менш, є шляхи для обміну даними з іншими додатками і для доступу до системних послуг:

— Можна розділити Linux Id між двома додатками, щоб вони були в змозі отримати доступ до файлів один одного. Щоб заощадити системні ресурси, додатки з одним і тим же ідентифікатором можуть запускатись в одному процесі й ділити між собою одну й ту ж віртуальну машину (англ. Virtual Machine, VM).

— Додаток може запитати дозволу на доступ до персональних даних пристрою, таких як контакти користувача, SMS повідомлень, розділ монтування (SD карту), камеру, Bluetooth і багато іншого. Всі дозволи програмі мають бути надані користувачем під час встановлення. При розробці всі привілеї, які використовує додаток декларуються в маніфест файлі. Маніфест файл – це файл в який інкапсулюється вся архітектура додатку. Він має розширення XML. Наприклад, щоб надати додатку дозвіл на використання Internet, потрібно в маніфесті написати наступні рядки коду:

```
<uses-permission android:name="android.permission.INTERNET" />
```

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Бібліотека Volley це HTTP бібліотека, яка робить мережевий обмін Android додатків більш легким і, найголовніше, більш швидким. Volley доступний для використання через відкрите AOSP сховище.

Volley надає наступні переваги:

- автоматичне планування мережевих запитів;
- кілька паралельних мережевих підключень;
- прозоре кешування на диску і в пам'яті відповідей, використовуючи стандартну

HTTP когерентність кеш;

- підтримка пріоритетів запитів;
- API для скасування запиту;
- простота настройки, наприклад, для повторних спроб і відстрочки;
- строгий порядок, який дозволяє легко і правильно заповнити призначений для користувача інтерфейс даними отриманих з мережі асинхронно;
- інструменти для налагодження і трасування.

Volley виділяється серед RPC операцій, використовуваних для заповнення призначеного для користувача інтерфейсу, таких як отримання сторінки результатів пошуку у вигляді структурованих даних. Вона легко інтегрується з будь-яким протоколами і поставляється з підтримкою сирих рядків, зображень і JSON. Надаючи вбудовану підтримку функції, які вам необхідні, Volley звільняє від написання стандартного коду і дозволяє зосередитися на логіці, яка специфічна для застосування.

Volley не підходить для скачування великих даних або потокових операцій, так як Volley зберігає всі відповіді в пам'яті під час аналізу і розбору даних.

Ядро бібліотеки Volley розробляється у відкритому AOSP сховище в frameworks / volley і містить основу диспетчеризації запитів, а також набір найбільш часто вживаних утиліт, доступних в папці "toolbox". Найпростіший спосіб додати Volley в проект це клонувати репозиторій Volley, і встановити її в якості бібліотеки проекту:

Git клонує репозиторій, використовуючи наступну команду в командному рядку:

```
git clone https://android.googlesource.com/platform/frameworks/volley
```

Після чого імпортують завантажені файли з вихідним кодом в додаток проекту в якості бібліотеки Android проекту.

Існує два основних класи:

— Request queue: використовується для відправки мережових запитів, можете створювати клас request queue, де хочете, але, як правило його створюють під час запуску і використовують як Singleton;

— Request: він містить всі необхідні деталі для створення виклику Web API. Наприклад: який метод використовувати (GET або POST), дані запиту, response listener, error listener.

Стандарт JSON є частиною стандарту ECMAScript починаючи з 1999 року, коли ECMA визначила функцію eval () сприймає формат. Він став набирати популярність разом з наростаючим успіхом Ajax. Слово JSON часто з'являється коли хто-небудь починає розмову про Ajax. Відомо, що JSON є іншим форматом даних, що він заміщає XML, і що багато програмісти активно підтримують його.

Перевага JSON полягає в тому, що він розпізнається в JavaScript. Немає необхідності розбирати документ, як це відбувається з XML, для передачі даних через інтернет.

Переваги JSON:

–Простий для читання і розуміння.

–Простий для використання.

–Має можливості розширення

І XML і JSON мають можливість інтегрувати великий обсяг даних в бінарній формі. Файл JSON дозволяє завантажувати дані з сервера або на сервер. Наприклад, збереження вмісту форми, яка була тільки що заповнена користувачем. Процес включає три фази: обробку браузером, обробку сервером, і обмін даними між ними.

Файл JSON використовується різними програмними мовами, включаючи PHP і Java, завдяки наявності парсерів, які дозволяють отримувати вміст і навіть конвертувати його в класи і атрибути мови. На сайті JSON можна знайти парсери для багатьох мов програмування [4].

Зокрема, бібліотека Volley містить в собі JsonObjectRequest, що слугує для відправки запитів на віддалений сервер через URL-адресу. Синтаксис метода GET , що використовувався в нашому додатку приведено нижче:

```
JsonObjectRequest jsonObjectRequest = new JsonObjectRequest (Request.Method.GET,
urlFinanceUajs,null,
    new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject s) {
            try {
            }catch (JSONException e){}
            try {}catch (JSONException ex){
                ex.printStackTrace();
                Toast.makeText(getApplicationContext(),"Error: connection to server failed! Try again later",
Toast.LENGTH_LONG).show();
            }
        }, new Response.ErrorListener() {
    }
}
```

Для розробки мобільних додатків під Android необхідним є досить широкий інструментарій розробника. Він включає в себе необхідні бібліотеки, від Java, Android SDK, середовище програмування, а також набір емуляторів, програм налагоджувачів, різноманітних екранів стану операційної системи та систем моніторингу та логування.

Можна виділити необхідні інструменти, без яких розробка мобільних додатків під Android просто неможлива. З іншого боку, існує велика кількість допоміжних систем, що в деякій мірі спрощують процес розробки.

До обов'язкових інструментів відноситься *Android SDK* - набір засобів програмування, який містить інструменти, необхідні для створення, компіляції і збірки мобільного додатку [9].

Розглянемо коротко найбільш важливі інструменти, що входять до складу Android SDK:

SDK Manager - інструмент, що дозволяє завантажити компоненти Android SDK. Він відображає пакети Android SDK і їх статус: встановлений (Installed), не встановлено (Not Installed), доступні оновлення (Update available).

Debug Monitor - самостійний інструмент, що надає графічний інтерфейс для декількох інструментів, призначених для аналізу та налагодження Android додатків:

- *DDMS (Dalvik Debug Monitor Server)* - надає послуги перекидання портів, захоплення екрану пристрою, інформацію про потоки і динамічну пам'ять пристрою, виведення інформації про дії Android в реальному часі (logcat) і багато іншого.

- *Hierarchy Viewer* дозволяє налагоджувати і оптимізувати користувацький інтерфейс Android програми.

- *Tracer for OpenGL ES* - інструмент для аналізу OpenGL | ES коду, що використовується в мобільному додатку дозволяє захоплювати команди OpenGL | ES і демонструвати їх окремими кадрами, що допомагає зрозуміти як виконуються графічні команди.

- *Android Emulator (emulator)* - віртуальний мобільний пристрій, який створюється і працює на комп'ютері розробника. Він використовується для розробки і тестування мобільних додатків без залучення реальних пристроїв.

- *AVD Manager* - надає графічний інтерфейс для створення віртуальних Android пристроїв (AVDs), передбачених Android Emulator, і управління ними.

Android Debug Bridge (adb) - гнучкий інструмент, що дозволяє керувати станом емулятора або реального Android пристрою, підключеного до комп'ютера. Також може використовуватися для установки Android програми (.apk файл) на реальний пристрій [5].

Висновки й перспективи подальших досліджень. В результаті дослідження була створена програма для Android, яка взаємодіє з віддаленим сервером, та обмінюється з ним інформацією у *GET*-запитах. Запити відбуваються у єдиному потоці. У разі виникнення помилок програма повідомляє про це користувача за допомогою впливаючих повідомлень. Програма протестована на планшетному комп'ютері Ainol Novo 9 Spark (Android 4.3), смартфоні Sony Ericsson Xperia c2305 (Android 5.1) та ще одному смартфоні Lenovo p780 (Android 4.2). На всіх пристроях програма коректно встановлювалась та працювала. За рахунок використаних алгоритмів збереження мережного трафіку, програма швидко отримувала потрібну інформацію від серверу навіть при повільній швидкості підключення.

В нашій статті ми розглянули операційну систему Android, розглянули процес моделювання роботи Android-додатку перегляду курсів валют, організували збір статистичних даних з банків, розглянули механізм передачі даних через мережу Інтернет, розглянули і дослідили роботу стандарту передачі даних JSON.

У програмі може бути покращений інтерфейс користувача - можна додати більше функції у меню. Також можна максимально ефективно використовувати екран якщо використовується пристрій з великим екраном. Для цього потрібно додати декілька макетів та задати їм відповідну поведінку в залежності від розміру екрану. Також замість стандартних макетів можна використовувати фрагментацію інтерфейсу. Ця технологія дозволяє більш гнучко керувати розміщенням елементів на екрані користувача.

Для збільшення швидкості роботи у програму можна додати кешування. Кешування у Android відбувається за допомогою запису потрібної інформації до бази даних *SQLite*. *SQLite* - база даних, яка для зберігання інформації використовує текстові файли.

Вони будуть кожний раз завантажуватися з серверу у *JSON* - масиві, після чого користувач зможе їх переглянути.

1. Дейтел П., Дейтел Х., Дейтел Э., Моргано М. Android для програмістів. Створюємо програми на Android. «Питер». – 2012.
2. Комагинени С., Маклин Д., Хашими С. Розробка програм для Android. «Питер». – 2011.
3. Медник З., Дорнин Л. Программирование под Android. Издательство Питер, 2012.
4. Роджерс Р., Ломбардо Д. Android. Разработка приложений. — М.: ЭКОМ Паблшерз,
5. Фелкер Д. Android: разработка приложений для чайников. - М.: Диалектика, 2012.
6. Х. М. Дейтел, П. Дж. Дейтел, С. И. Сантри Програмування на Java - ДияСофтЮП, 2010.