

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НУ «Запорізька політехніка»**

Факультет комп'ютерних наук та технологій  
Кафедра «Комп'ютерні системи та мережі»

**РУДЕНКО ВІТАЛІЙ ВОЛОДИМИРОВИЧ**  
Група КНТ-612м

**КОМП'ЮТЕРНА СИСТЕМА ДИСТАНЦІЙНОГО ЗБИРАННЯ**  
**ANDROID-ДОДАТКІВ**

**АВТОРЕФЕРАТ**  
магістерської роботи на здобуття  
освітньо-кваліфікаційного рівня «магістр»  
за спеціальністю 123 «Комп'ютерна інженерія»  
освітньої програми «Спеціалізовані комп'ютерні системи»

2023 р.

Магістерська робота є рукопис.

Робота виконана в національному університеті "Запорізька політехніка", на кафедрі комп'ютерних систем та мереж

**Керівник**

кандидат технічних наук, доцент  
Скрупський Степан Юрійович,

**Рецензент:**

СТЕПАНЕНКО Олександр Олексійович  
к.т.н.,доцента кафедри «Програмні засоби»  
НУ «Запорізька політехніка»

Захист відбудеться " 19 " грудня 2023р.

Секретар екзаменаційної комісії, старший викладач  
комп'ютерних систем та мереж Н.А. Куликовська

## **ЗАГАЛЬНА ХАРАКТЕРИСТИКА ДОСЛІДЖЕННЯ**

**Актуальність.** В епоху, яка визначається постійно прискореними темпами технологічного прогресу та дедалі більшою розповсюдженістю мобільних пристроїв, розробка додатків для Android стала однією з основних сфер розробки програмного забезпечення. Оскільки вимоги до інноваційних та ефективних додатків продовжують зростати, також зростає потреба в нових парадигмах розробки, які можуть оптимізувати процес розробки додатків і оптимізувати використання ресурсів. У цьому контексті концепція створення додатків Android на віддаленому комп'ютері набула значного інтересу, пропонуючи багатообіцяючий шлях для вирішення проблем і можливостей, які представляє ця динамічна екосистема.

Ця дипломна робота досліджує критичну важливість і невід'ємну оптимальність збірки додатків Android на віддалених машинах, з'ясовуючи переваги, проблеми та наслідки цієї зміни парадигми у сфері розробки мобільних додатків.

**Мета та завдання дослідження.** Мета дипломного проекту — створення системи для дистанційної збірки Android-додатків, що забезпечує ефективну, гнучку та зручну роботу розробників. Поставлена мета була досягнута шляхом вирішення наступних завдань:

- детальний аналіз існуючих систем збирання Android-додатків для визначення їх переваг і недоліків;
- розробка власної дистанційної системи збирання Android-додатків, використовуючи мову програмування Kotlin та інструментарій Android Studio;
- розгляд варіантів архітектур Android-додатків та вибір оптимального підходу до розробки;
- аналіз програмного та апаратного забезпечення для ефективної роботи машини для збирання додатків;
- порівняння розробленої системи з існуючими аналогами для визначення її переваг та можливостей.

**Об'єкт дослідження** Об'єкт дослідження – це комп'ютерна система дистанційного збирання даних, яка базується на Android-додатках.

**Предмет дослідження** Предмет дослідження – системи збирання Android-додатків, та їх потенціальні недоліки.

**Методи дослідження** У цій роботі детальний аналіз, вибір технологій (мова програмування Kotlin, IDE Android Studio), аналіз

різних архітектур додатків, вибір програмного та апаратного забезпечення, а також оцінку переваг розробленої системи порівняно з існуючими аналогами.

**Наукова новизна отриманих результатів:** полягає у запропонованому підході, заснованому на використанні власної дистанційної системи збирання Android-додатків, що має можливість збирання додатків різної архітектури з різною кількістю модулів, комплексне рішення, яке відповідає різноманітним вимогам розробки програм для Android. **Практичне значення отриманих результатів:** розроблено систему, використання якої підвищило ефективність розробки та роботи Android-додатків в порівнянні з аналогами.

## ОСНОВНИЙ ЗМІСТ РОБОТИ

**У першому розділі** виконується огляд сучасних систем збирання додатків, які можуть бути використані в Android-розробці. Розглядаються переваги та недоліки кожного з методів.

Локальна збірка програм для Android починається з налаштування середовища розробки на локальній машині розробника. Під час створення проекту вони можуть налаштувати різні аспекти програми, такі як її назва, ім'я пакета та мінімальний рівень Android API, який він підтримуватиме. Крім того, розробники можуть вибирати між різними шаблонами проектів, які визначають початкову структуру програми.

Дистанційне збирання Android-додатків забезпечує альтернативний підхід до процесу розробки. Цей метод особливо корисний у середовищах спільної роботи та під час вирішення ресурсомістких завдань.

Віддалена збірка Android-додатків є процесом складання додатків на сервері або віддаленій машині, а не на локальній машині розробника. Це процес, при якому розробник відправляє вихідний код та ресурси на віддалений сервер, де складання програми виконується на віддаленій машині. Результати збирання можуть бути передані назад на локальну машину розробника.

**У другому розділі** вибір відповідного програмного забезпечення при розробці та збиранні додатків Android. Розглядається різні аспекти цього вибору, включаючи мови програмування та інші ключові компоненти.

Java-це класна об'єктно-орієнтована мова програмування високого рівня, розроблена для того, щоб зробити ваш додаток максимально залежним. Це означає, що скомпільований код Java може виконуватися на будь-якій платформі з підтримкою Java без перекомпіляції.

Kotlin – це крос платформна, статично набрана, універсальна мова програмування високого рівня з виведенням типів. Kotlin розроблений для повної взаємодії з Java, а JVM-версія стандартної бібліотеки Kotlin залежить від бібліотеки класів Java, яка простіша за синтаксисом через визначення типу Kotlin в першу чергу призначений для JVM, але не призначений для машинного коду через JavaScript або LLVM.

Порівняння Java та Kotlin для розробки віддаленого збирання Android-додатків є важливим аспектом вибору технології в контексті проекту. Обидві мовні платформи пропонують низку переваг, але мають свої особливості.

Перевага Kotlin перед Java для розробки Android-додатків можуть ґрунтуватися на кількох ключових аспектах, враховуючи вимоги проекту та переваги команди розробників:

- сучасний синтаксис і виразність;
- безпека та нульові значення за промовчанням;
- робота з асинхронним кодом;
- повна сумісність з Java;
- активна підтримка Google.

Використовуймо операційну систему Ubuntu яка поєднує в собі відкритість і безкоштовність Linux зі зручністю і функціональністю. Це може бути оптимальним вибором для розробників, особливо в контексті Android-додатків. Ubuntu постає як надійна та добре підходяща операційна система для спеціальної машини, на яку поставлено завдання дистанційного створення додатків Android. Його сумісність із інструментами розробки Android, відданість безпеці, активна підтримка спільноти та бездоганна інтеграція роблять його оптимальним вибором.

Вибір Android Studio як середовище розробки для нашого проекту обумовлений не лише широкою функціональністю та зручністю використання, а й вільним доступом для всіх розробників. Це інтегроване середовище розробки надає все необхідне для того, щоб з легкістю створювати та вдосконалювати Android-додатки, роблячи його оптимальним та незрівнянним вибором для успішного втілення наших розробних завдань.

Таблиця 1 - Апаратне обладнання

|                    |                                                             |
|--------------------|-------------------------------------------------------------|
| Материнська плата  | Gigabyte Z790 UD                                            |
| Процесор           | Intel Core i7-14700K BOX, 16 ядер по 3.4GHz                 |
| Відеоадаптер       | Інтегрований на материнській платі                          |
| Оперативна пам'ять | 32 DDR5 5600 МГц                                            |
| Накопичувач        | Kingston KC3000 SKC3000D 2 TB                               |
| Блок живлення      | Chieftec Polaris 1050 Вт                                    |
| Операційна система | Ubuntu                                                      |
| Клавіатура/миша    | Клавіатура PS/2, миша USB                                   |
| Мережевий адаптер  | Інтегрований адаптер Gigabit Ethernet на материнській платі |
| Монітор            | «32» LG 32MN500M, 1920x1080 (16:9)                          |

**У третьому розділі** проектування налаштованої системи, призначеної для бездоганної інтеграції в сучасні робочі процеси розробки Android.

Для дослідження системи було взято вихідний код одного реального багатомодульного проєкту. На його основі було сформовано три варіанти:

- багато модулів;
- менше модулів;
- лише один модуль.

Порівнявши продуктивність і характеристики кожної конфігурації буде визначено оптимальний баланс між модульністю та монолітністю додатку.

Базовий проєкт має 59 модулів, які складаються з 864 файлів Kotlin і 1 файлу Java. Крім того, він включає 65 файлів активів, 1 файл прототипу та 161 файл ресурсів.

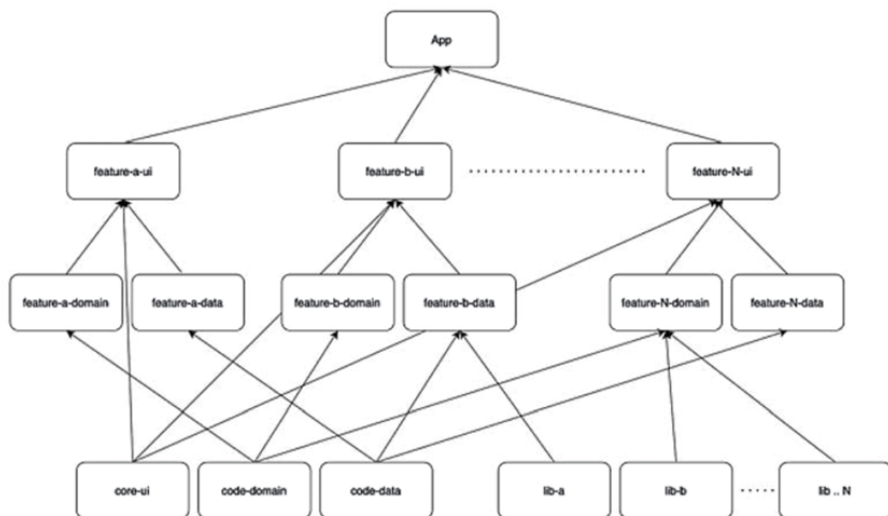


Рисунок 1 – Проєкт із 59 модулів

Перший етап - тестування базового проєкту. Далі була зроблена його копія та об'єднані під модулі функцій в один модуль. Наприклад, функція А має модулі даних, домену та інтерфейсу користувача. Було зроблено переміщення даних та класів домену в модуль інтерфейсу користувача, що дозволило отримати один модуль замість 3. Користуючись таким алгоритмом вдалось зменшити кількість модулів до 33.

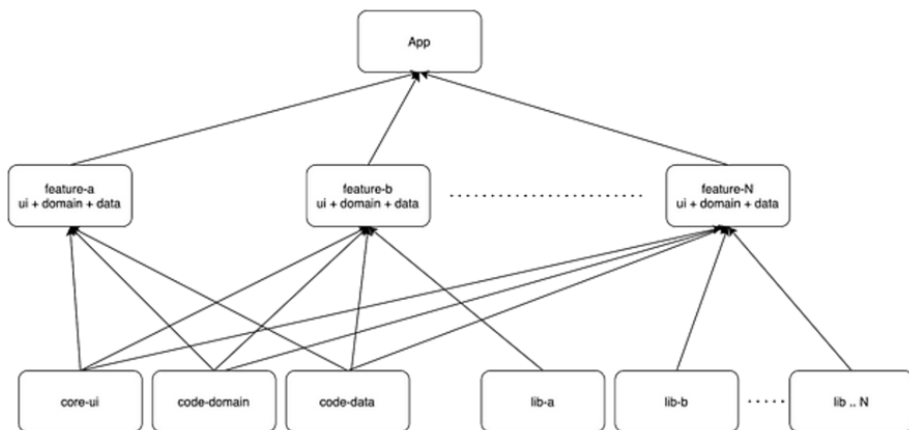


Рисунок 2 – Проєкт із 33 модулів

Останній етап - об'єднуємо всі модулі в один і отримуємо проєкт з монолітною структурою.

Таблиця 2 - Результати тестування операцій

| Показники       | Операція<br>assembleDevDebug |           |            | Операція<br>assembleDevRelease |           |            | Операція<br>assembleDevDebug без<br>очищення |           |            |
|-----------------|------------------------------|-----------|------------|--------------------------------|-----------|------------|----------------------------------------------|-----------|------------|
|                 | 1<br>модуль                  | 33-модуля | 59-модулів | 1<br>модуль                    | 33-модуля | 59-модулів | 1<br>модуль                                  | 33-модуля | 59-модулів |
| 1 запуск<br>(с) | 80                           | 98        | 102        | 185                            | 261       | 331        | 2                                            | 7         | 13         |
| 2 запуск<br>(с) | 74                           | 95        | 104        | 174                            | 229       | 249        | 2                                            | 7         | 11         |
| 3 запуск<br>(с) | 74                           | 94        | 102        | 194                            | 238       | 260        | 2                                            | 7         | 10         |
| 4 запуск<br>(с) | 75                           | 94        | 106        | 216                            | 268       | 283        | 2                                            | 7         | 11         |

Продовження таблиці 2

| Показники        | Операція<br>assembleDevDebug |           |            | Операція<br>assembleDevRelease |           |            | Операція<br>assembleDevDebug без<br>очищення |           |            |
|------------------|------------------------------|-----------|------------|--------------------------------|-----------|------------|----------------------------------------------|-----------|------------|
|                  | 1<br>модуль                  | 33-модуля | 59-модулів | 1<br>модуль                    | 33-модуля | 59-модулів | 1<br>модуль                                  | 33-модуля | 59-модулів |
| 4 запуск (с)     | 75                           | 94        | 106        | 216                            | 268       | 283        | 2                                            | 7         | 11         |
| 5 запуск (с)     | 74                           | 94        | 104        | 225                            | 336       | 368        | 2                                            | 7         | 10         |
| Середнє значення | 75,4                         | 95        | 103,6      | 198,8                          | 266,4     | 298,2      | 2                                            | 7         | 11         |
| Дельта (с)       | 0                            | 19,6      | 28,2       | 0                              | 67,6      | 99,4       | 0                                            | 5         | 9          |
| Дельта (%)       | 0                            | 26        | 37         | 0                              | 34        | 50         | 0                                            | 250       | 450        |

Результати даного експерименту показали, що монолітна архітектура продемонструвала найменший час збирання, тоді як підхід із багатьма модулями призвів до довшої тривалості компіляції. Загалом експеримент дав цінну інформацію про компроміси між монолітною та модульною архітектурами, підкреслюючи важливість вибору найбільш прийнятного підходу на основі вимог до проекту та бажаних результатів.

У випадку експерименту з тестуванням різних підходів, проект з монолітною архітектурою демонстрував найкращі результати, з найменшим часом збирання порівняно з багатомодульним проектом. Таким чином, дана робота надає цінну інформацію про компроміси між монолітною та модульною архітектурами та підкреслює важливість вибору оптимального підходу залежно від вимог до проекту та бажаних результатів.

## ВИСНОВКИ

У рамках дослідження було проведено детальний аналіз обраної сфери, де основною виявленою проблемою стала відсутність якісних та зручних систем дистанційного збирання Android-додатків. Як потенційне рішення цього питання та для підвищення користувацького досвіду було запропоновано розробити власну систему, враховуючи сучасні потреби розробників і недоліки аналогів таких систем.

Далі у роботі було здійснено глибоке дослідження основ систем збирання Android-додатків, розглянуті варіанти локальної та дистанційної збірки. Аналіз підтвердив, що дистанційні системи збирання додатків мають великий потенціал у вирішенні проблем розробки Android-додатків.

Для розробки системи обрано мову програмування Kotlin, яка є сучасною і у порівнянні з Java має багато переваг. Розробка проводилась в IDE Android Studio, яка не має аналогів і як найкраще підходить для розробки Android-додатків та роботи з Kotlin.

Було детально проаналізовано і обрано програмне та апаратне забезпечення для окремо машини для збирання додатків.

Акцентувалась увага на різні варіанти архітектури Android-додатків, детально розглядались переваги та недоліки кожного варіанту.

Завдяки дослідженню стає важливість вибору архітектури додатку і слідкування за оптимальною кількістю модулів у наявному додатку з модульною архітектурою.

Розроблена система має ряд переваг, яких немає у аналогів, а саме:

- інтеграція графічного інтерфейсу користувача;
- можливість працювати з різними архітектурами;
- відкритість та безкоштовність;

Дозволило зрозуміти деякі переваги монолітної архітектури і важливість оптимізації кількості модулів.