

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

ЗВІТ

з лабораторної роботи №1
з дисципліни «Сучасні мобільні операційні системи»

Тема: «Activity – Робота з елементами екрану»

Варіант №4

Виконав:

студент 1 курсу, групи ІМ-51мн

Ковалев Олександр

Перевірив:

асистент, Нестерук Андрій Олександрович

Дата здачі: 10.02.2026

КИЇВ – 2026

Мета роботи. Знайомство з інтерфейсом середовища програмування та вивчення структури проекту.

Завдання. Встановити Java Development Kit та Android Software Development Kit. Створити проект програми, заповнити його дані, зберегти. Створити додаток з одним екраном (Activity). Створити два Activity та організувати переход між ними. Вміст Activity 1 – кнопка з ім’ям btn1. Вміст Activity 2 – TextView з текстом ”Параметр: значення_параметра”. Значення_параметра – з Activity 1. При запуску програми користувач повинен потрапляти на екран з Activity 1. Після натискання на кнопку btn1 необхідно здійснити переход до Activity 2 і передавати параметр з Activity 1. Як значення параметра використовувати своє прізвище.

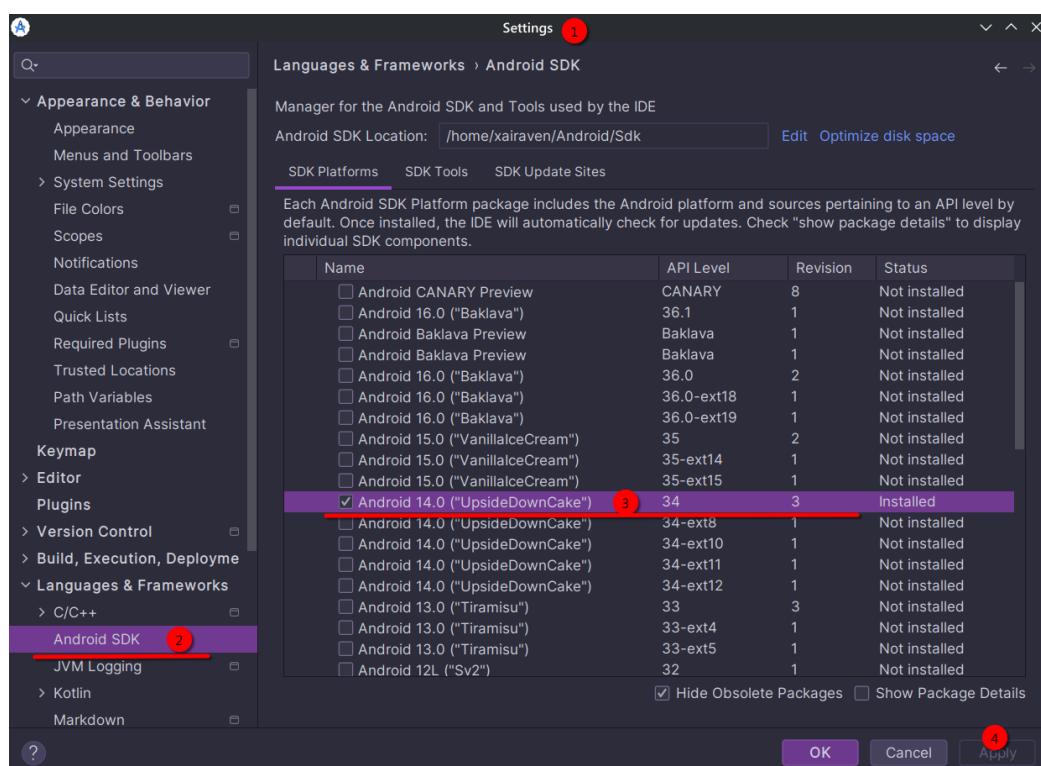
Хід роботи.

Для початку треба встановити JDK. Встановлено JDK версії 26.0:

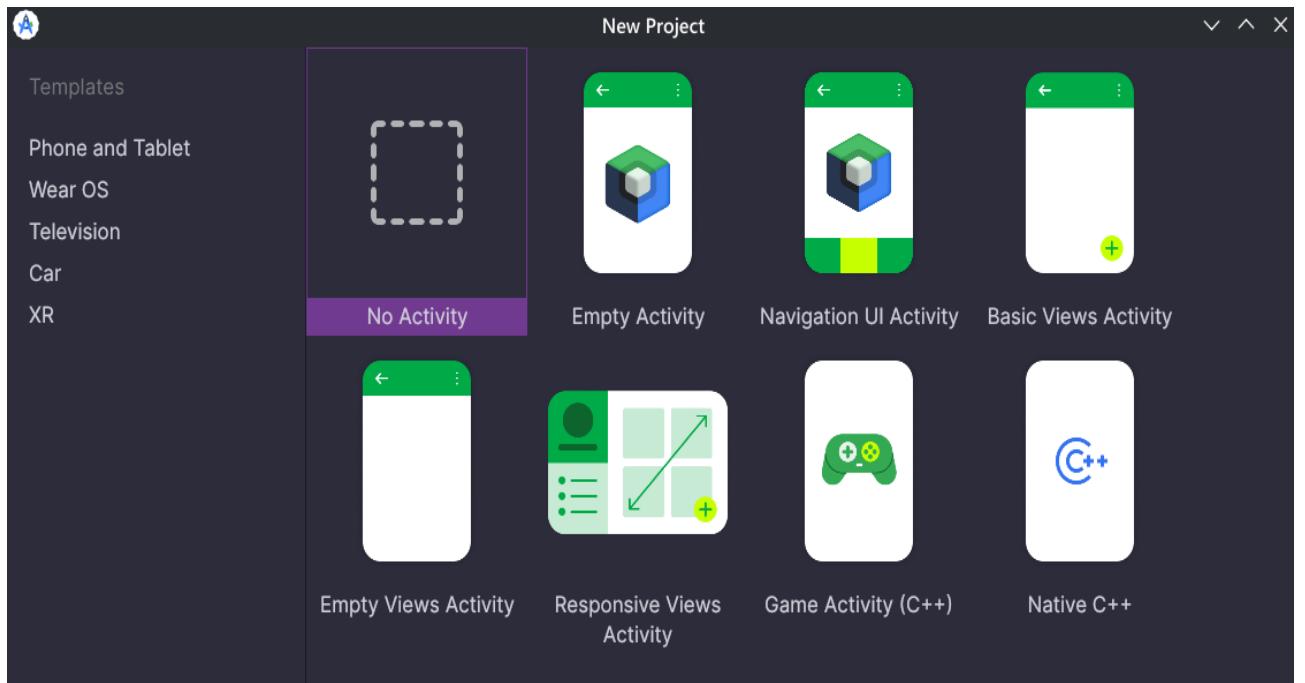
```
xairaven@laptop ~
$ sudo dnf install java-latest-openjdk.x86_64
Updating and loading repositories:
Repositories loaded.
Package "java-latest-openjdk-1:26.0.0.0.32-0.0.1.ea.fc43.x86_64" is already installed.

Nothing to do.
xairaven@laptop ~
$
```

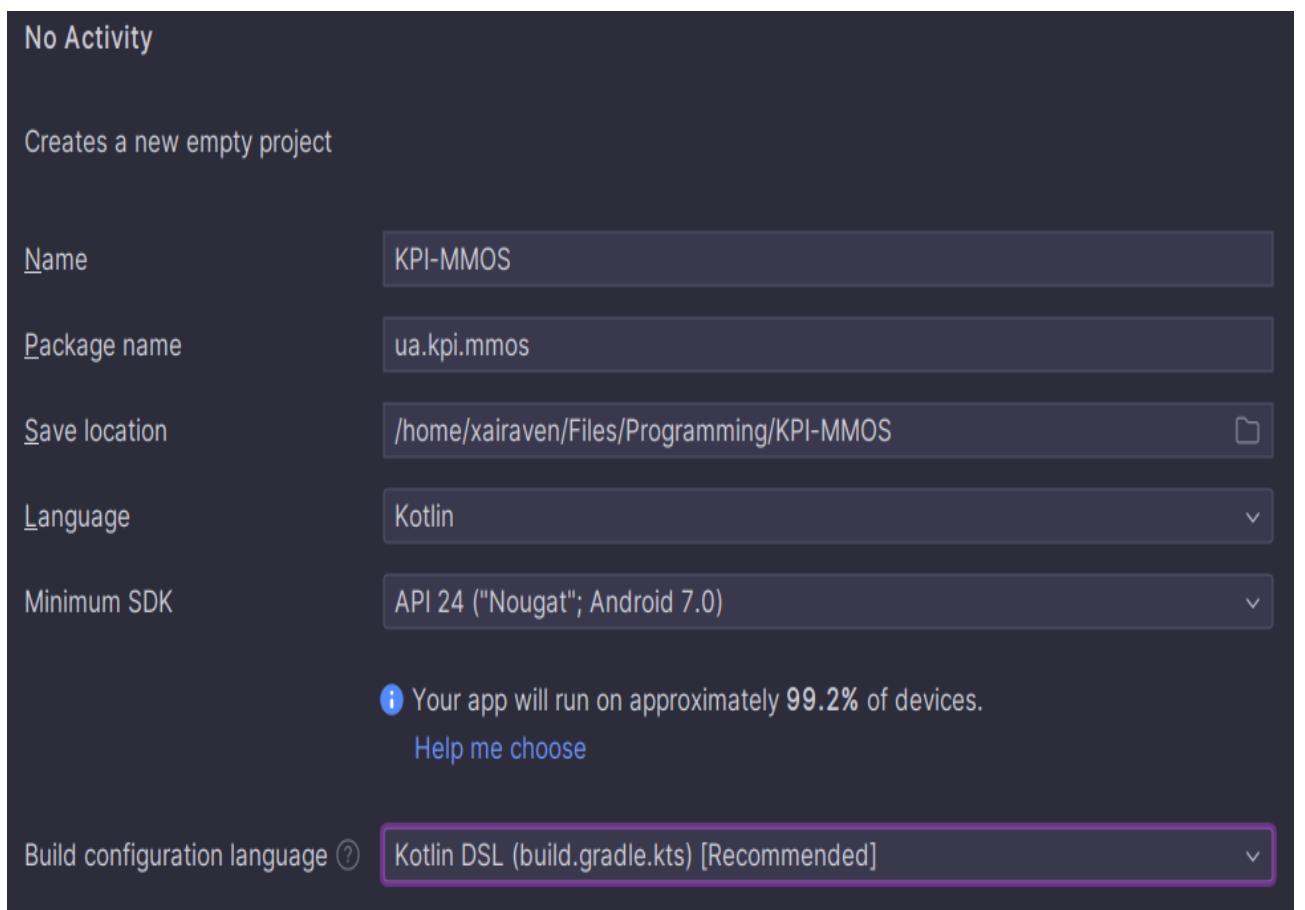
Далі, за пропозицією Android Studio, завантажено та встановлено Android SDK версії 14.0 (кодова назва «UpsideDownCake») з API Level 34:



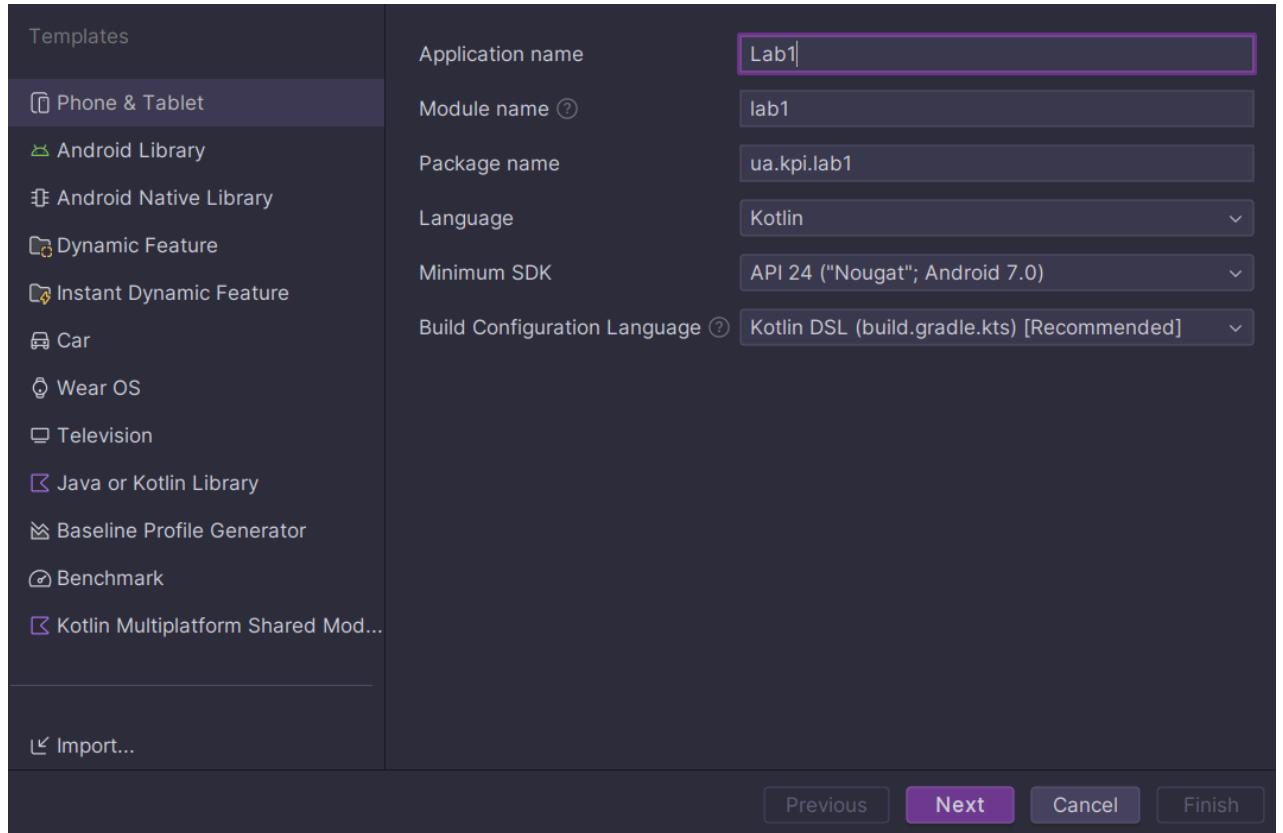
Створено новий проект. В якості шаблону обрано «No Activity», оскільки планується використовувати один кореневий каталог (репозиторій) для збереження проектів усіх лабораторних робіт у вигляді окремих модулів.



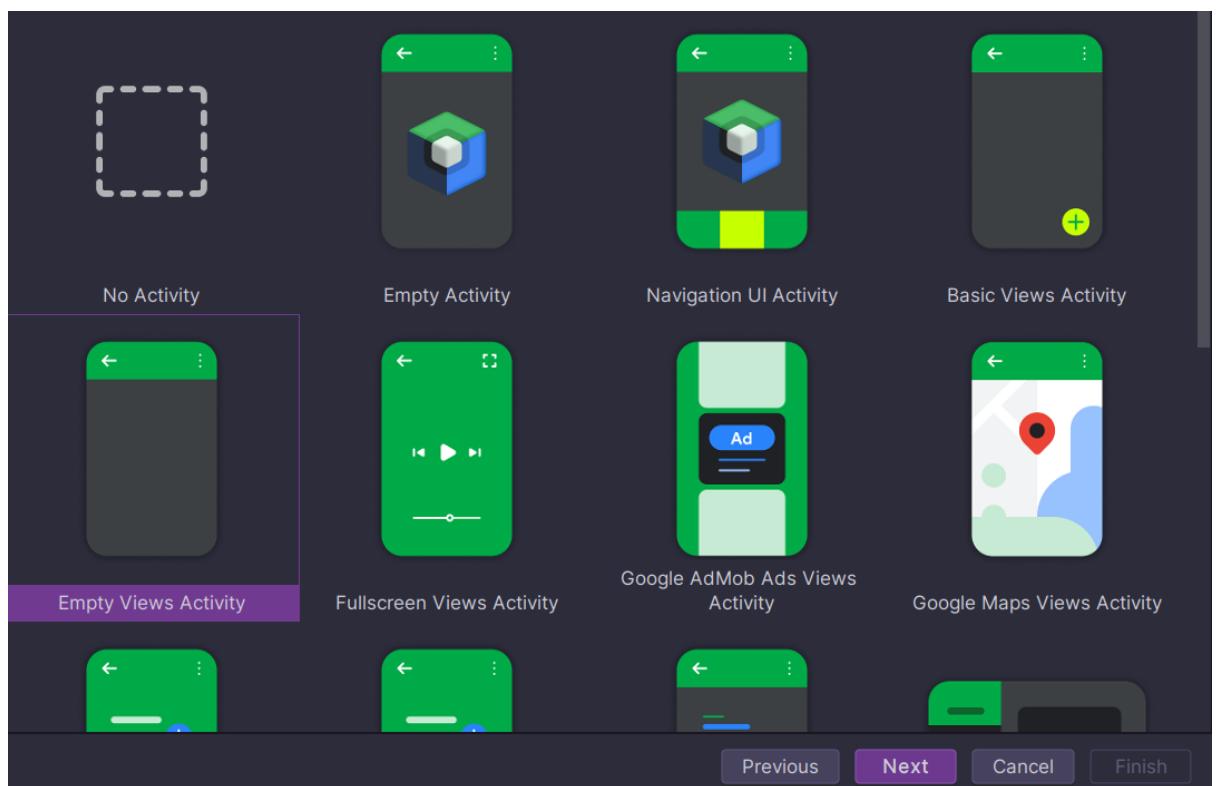
Заповнено метадані проекту. В якості основної мови розробки обрано Kotlin.



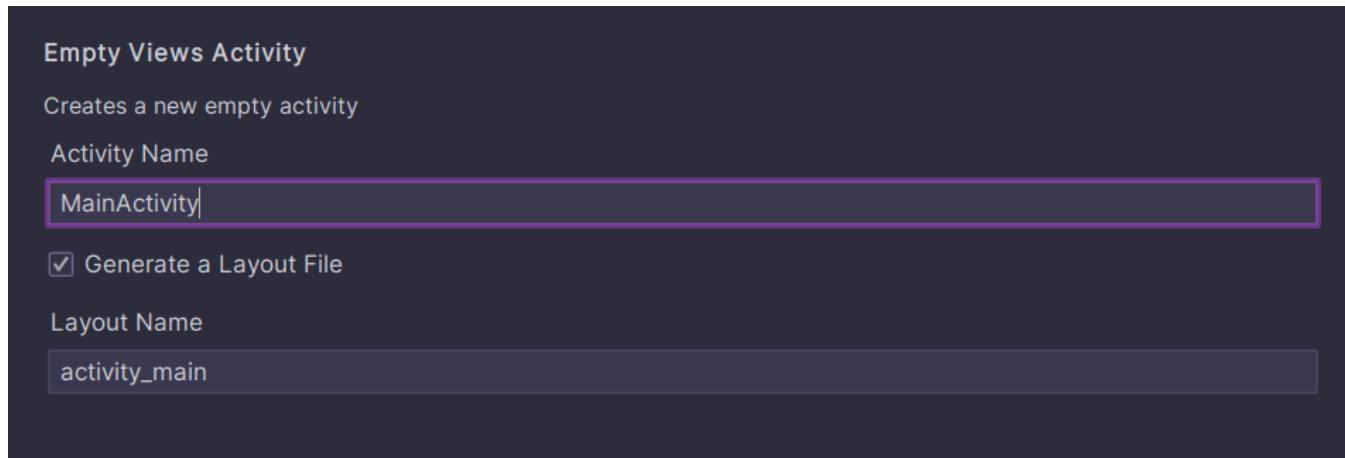
Після завершення індексації файлів сформовано структуру проекту. Кореневий каталог слугуватиме репозиторієм. Стандартний модуль `app` видалено, а замість нього створено новий модуль для поточної роботи під назвою `lab1`. Мінімальною версією SDK обрано API 24 (Android 7.0 «Nougat»).



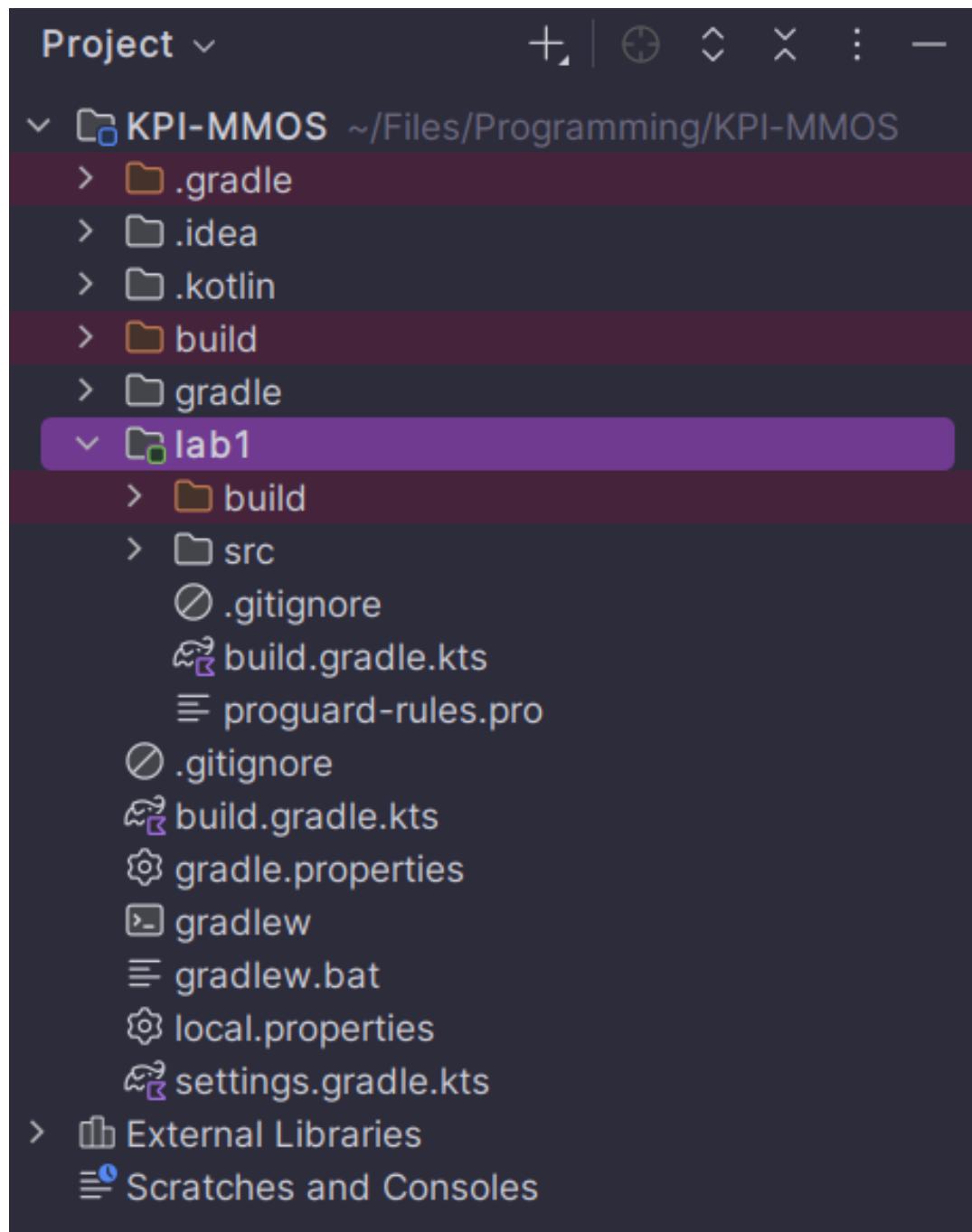
Для модуля `lab1` використано шаблон активності «Empty Views Activity»:



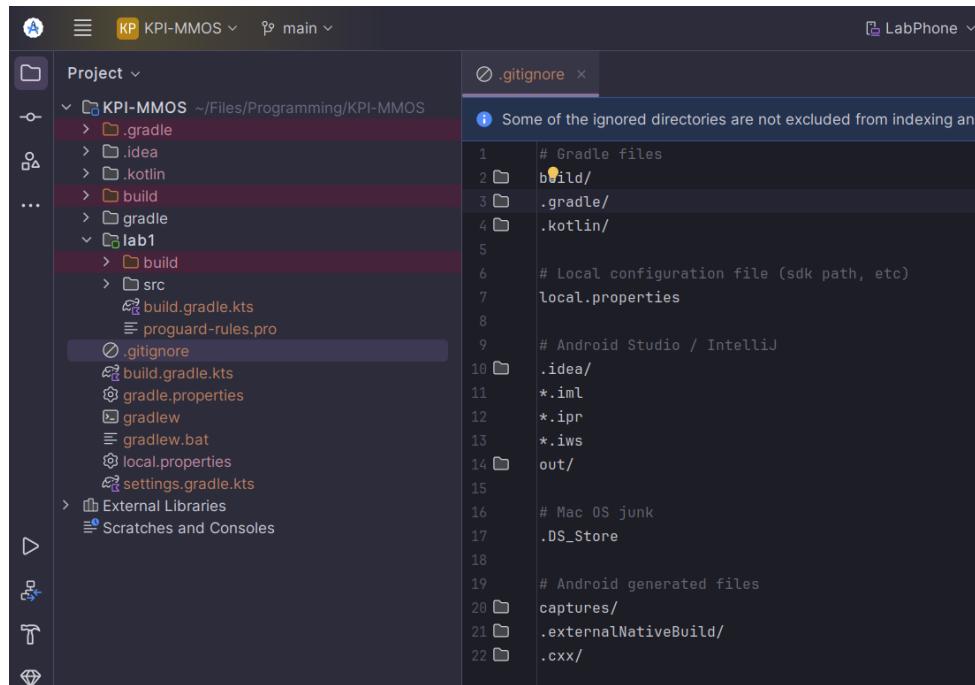
Ім'я головного класу активності (Activity) залишено за замовчуванням:



Шаблон проекту успішно згенеровано:



Для коректного версіонування проекту налаштовано файл `.gitignore`, щоб виключити з репозиторію тимчасові файли та файли збірки:



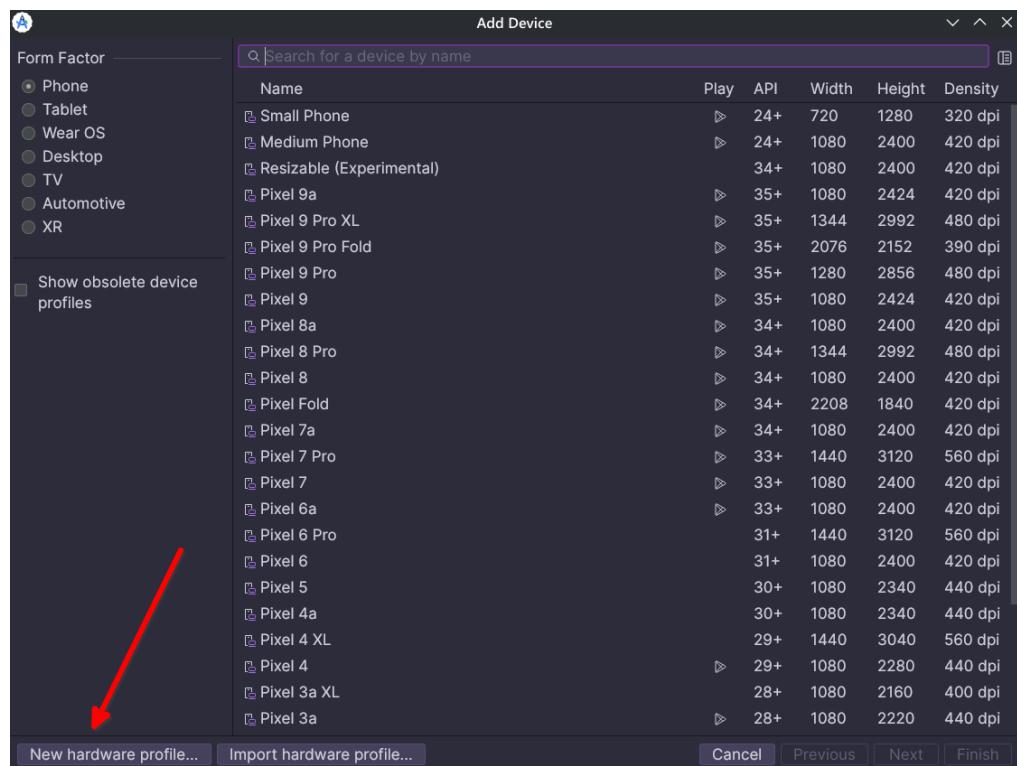
Виконано першу фіксацію змін (commit):

```
xairaven@laptop ~/Files/Programming/KPI-MMOS <main>
$ git init
Reinitialized existing Git repository in /home/xairaven/Files/Programming/KPI-MMOS/.git/
xairaven@laptop ~/Files/Programming/KPI-MMOS <main>
$ git add .
xairaven@laptop ~/Files/Programming/KPI-MMOS <main>
$ git commit -m "(Workspace) Init: Repository and Lab1 template"
[main (root-commit) 9273435] (Workspace) Init: Repository and Lab1 template
 34 files changed, 854 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 build.gradle.kts
 create mode 100644 gradle.properties
 create mode 100644 gradle/libs_versions.toml
```

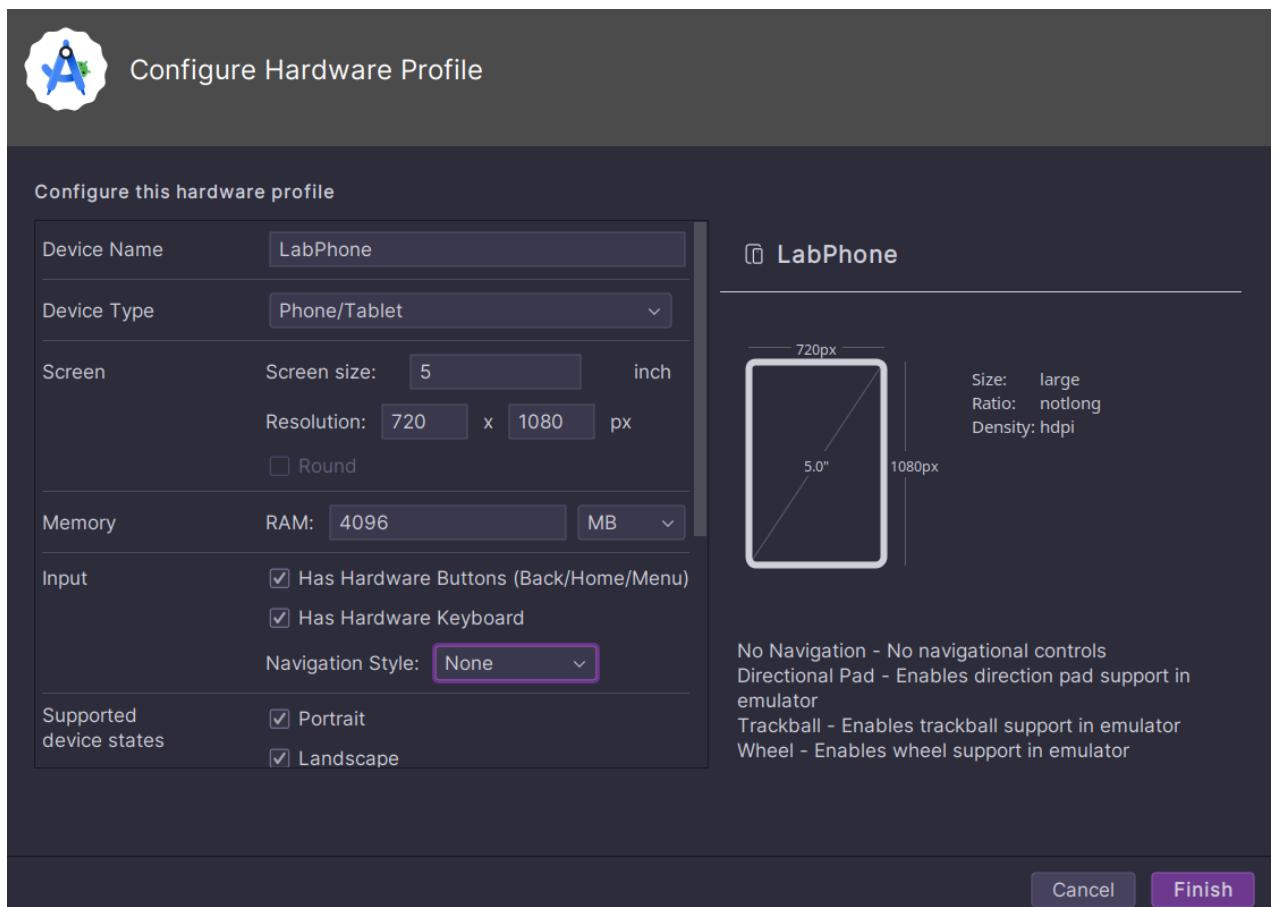
Проект завантажено на віддалений сервер GitHub у приватний репозиторій:

The GitHub interface shows the repository `KPI-MMOS` (Private). The repository was created by `xairaven` 4 minutes ago. It contains 65 files and 34 commits. The repository includes files like `gradle`, `lab1`, `.gitignore`, `build.gradle.kts`, `gradle.properties`, `gradlew`, `gradlew.bat`, and `settings.gradle.kts`. The `README` file is present but empty.

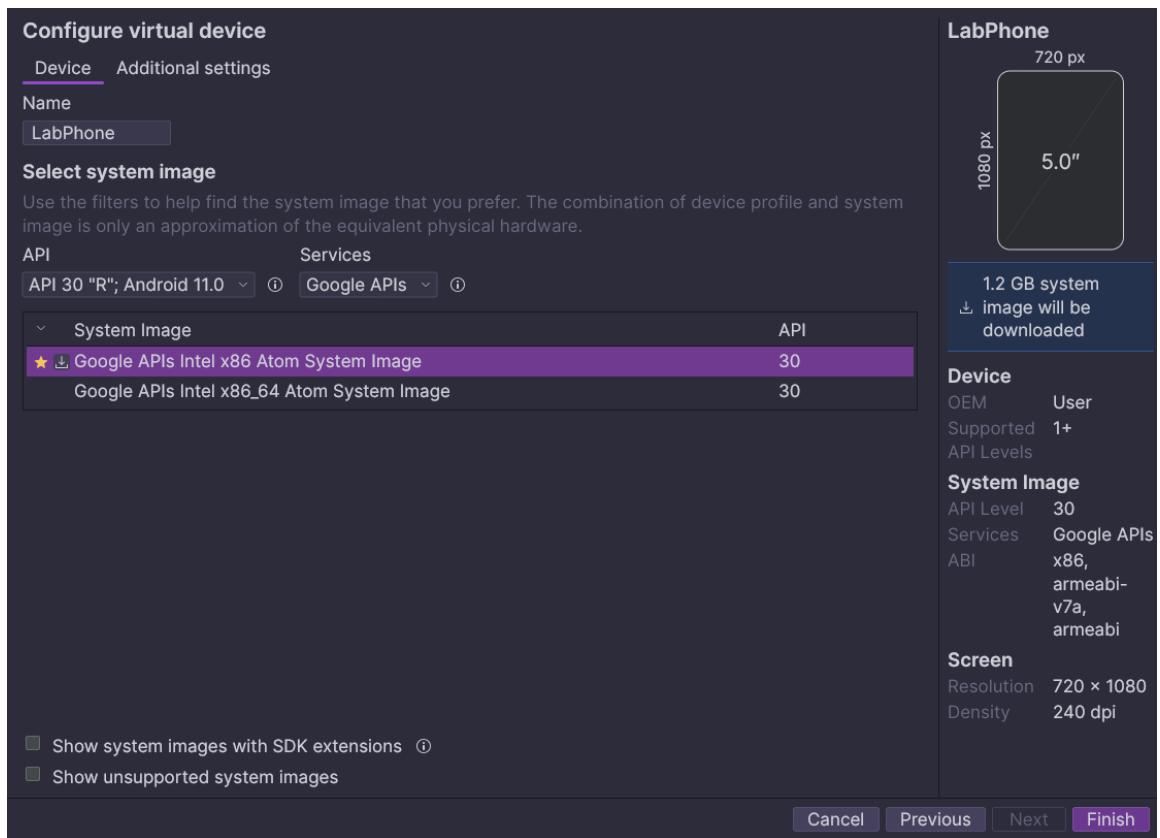
Для перевірки роботи додатку налаштовано емулятор (Android Virtual Device). Створено новий профіль пристрою:



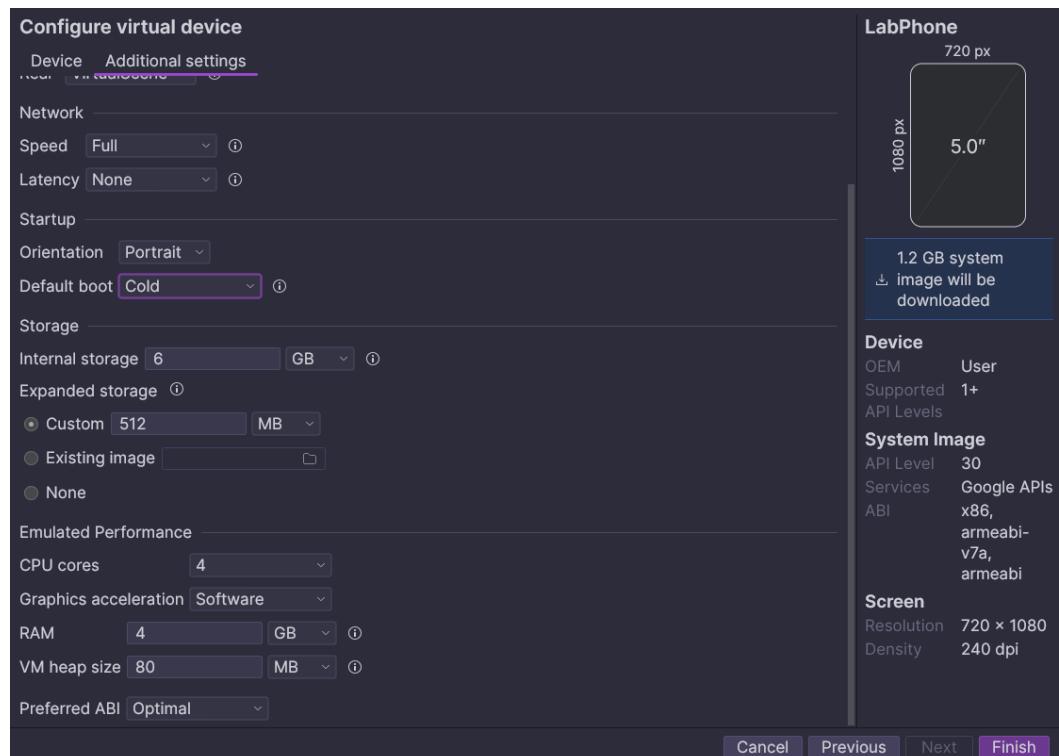
З метою оптимізації споживання ресурсів ПК, обрано пристрій з невеликою роздільною здатністю екрану (720p) та виділено 4 ГБ оперативної пам'яті.



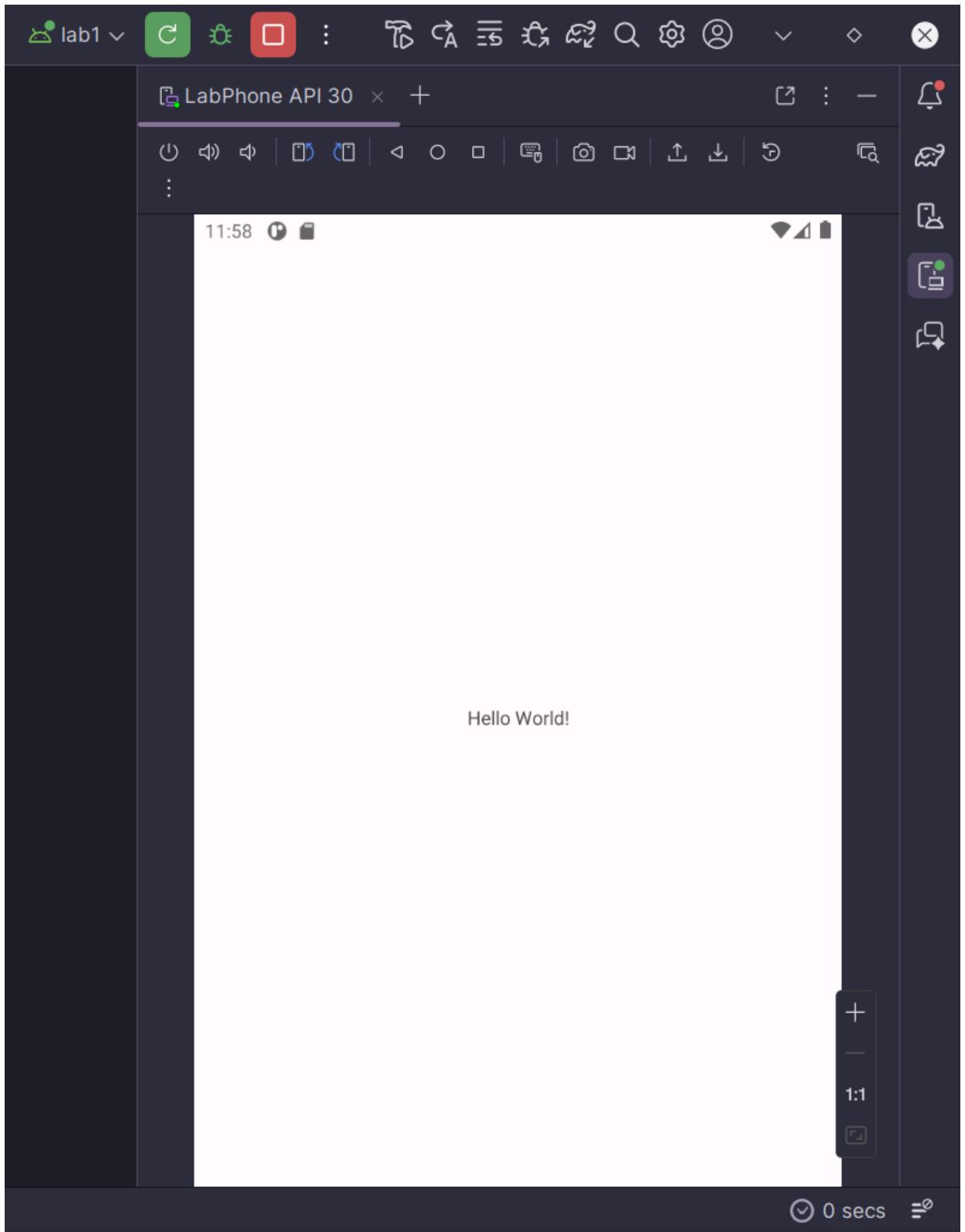
В якості операційної системи емулятора обрано Android 11 (API 30), яка є найбільш стабільною для емуляції в середовищі Linux.



Для покращення швидкодії емулятора увімкнено апаратне прискорення графіки (Hardware Acceleration). Тип завантаження встановлено на «Cold Boot» для уникнення помилок кешування стану.



Успішно запущено базовий шаблон додатку «Hello World»:



Розроблено графічний інтерфейс користувача (UI). Відредаговано XML-файли розмітки в каталозі `res/layout`. У головній активності (`MainActivity`) розміщено текстовий напис (`TextView`) та кнопку (`Button`). Елементам присвоєно ідентифікатори та налаштовано параметри прив'язки (`Constraints`) для коректного відображення.

Реалізовано логіку додатка в класі `MainActivity`. Оголошено змінні для кнопки та константи для передачі даних (прізвища). Додано слухач подій (Listener) на кнопку: при натисканні створюється об'єкт `Intent`, до якого додається прізвище студента за ключем `SURNAME_KEY`, після чого ініціюється запуск другої активності.

11:14 ① 📟

LTE 4G 🔋

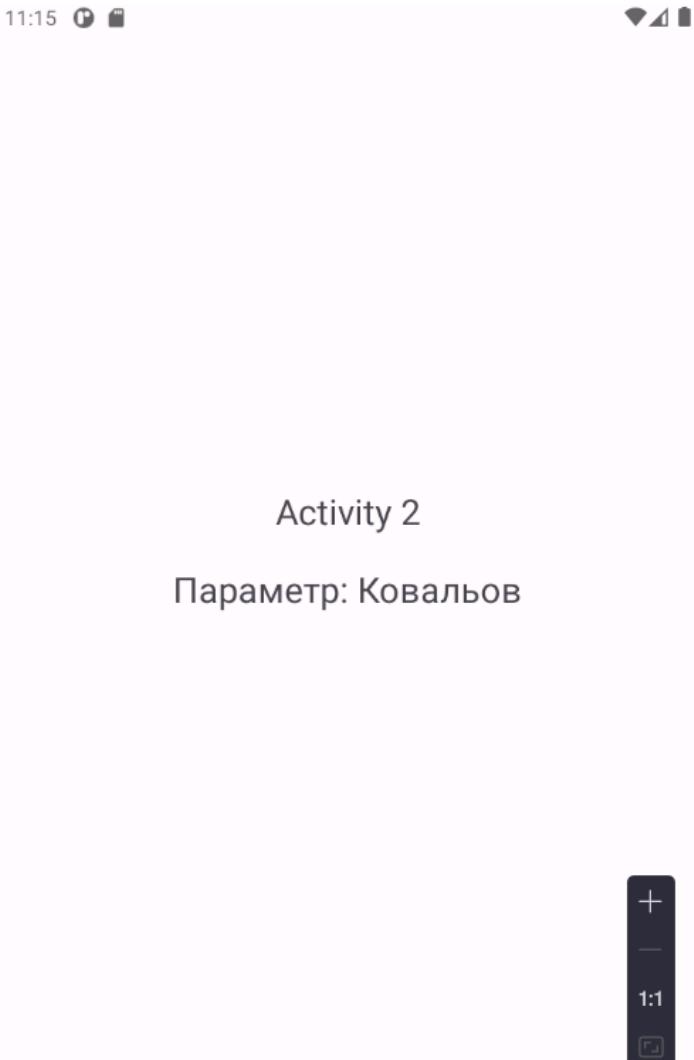
Activity 1

Передати прізвище



Для другої активності створено аналогічну розмітку, що містить текстове поле для виводу результату. У класі `SecondActivity` реалізовано отримання переданих даних з `Intent` та відображення їх на екрані за допомогою інтерполяції рядків.

Розмітка в другій активності аналогічна першій, окрім того, що там два написи. Їм також були присвоєні ідентифікатори, тощо. В логіці відбувається знаходження елементу напису, отримання результату за ключем з інтенту та присвоєння напису відповідного тексту, сформованого за допомогою інтерполяції.



Висновок. В ході виконання лабораторної роботи було встановлено та налаштовано необхідне програмне забезпечення для розробки мобільних додатків: Java Development Kit (JDK) та Android SDK. Було проведено ознайомлення з інтерфейсом середовища розробки Android Studio та вивчено структуру файлів Android-проекту.

Набуто практичних навичок у створенні та конфігурації віртуальних пристройів (AVD) для емуляції роботи додатків на ПК. Було розроблено додаток мовою Kotlin із використанням ConstraintLayout, який складається з двох екранів (Activity). Реалізовано логіку переходу між екранами та передачу даних (прізвища) з однієї активності в іншу за допомогою класу Intent.

Контрольні запитання.

1. Що таке мобільний додаток, мобільна платформа?

Мобільний додаток – це програмне забезпечення, спеціально розроблене для використання на портативних пристроях, таких як смартфони або планшети. Мобільна платформа – це операційна система, що керує апаратним забезпеченням мобільного пристрою та надає програмне середовище для виконання додатків (наприклад, Android або iOS).

2. Що собою являє архітектура мобільної платформи Android?

Архітектура Android побудована у вигляді багаторівневого стека програмного забезпечення. В її основі лежить ядро Linux, яке відповідає за драйвери та керування пам'яттю. Над ним знаходитьсья рівень апаратних абстракцій (HAL), нативні бібліотеки C/C++ та середовище виконання Android Runtime (ART). Вищим рівнем є Java API Framework, який надає інструменти розробникам, а на вершині знаходяться системні та користувацькі додатки.

3. Які основні компоненти Android-додатку?

До основних компонентів належать Activity (активність), яка відповідає за візуальний інтерфейс та взаємодію з користувачем, та Service (сервіс), що виконує фонові завдання без графічного інтерфейсу. Також важливими є Broadcast Receiver, який отримує та обробляє загальносистемні повідомлення, та Content Provider, що забезпечує керування даними та їх спільне використання між різними додатками.

4. Що собою являє структура Android-проекту?

Структура Android-проекту – це ієрархія каталогів та файлів, що розділяє логіку програми, ресурси та налаштування збірки. Вона зазвичай складається з модулів (наприклад, app), які містять вихідний код, файли маніфесту та ресурси, а також скриптів Gradle, які керують процесом компіляції та залежностями проекту.

5. Що містить файл конфігурації `AndroidManifest.xml`, папки `java` та `res`?

Файл `AndroidManifest.xml` описує фундаментальні характеристики додатку, такі як назва пакету, дозволи, зареєстровані компоненти (активності, сервіси). Папка `java` містить файли з вихідним кодом програми (Kotlin або Java), де прописана логіка роботи. Папка `res` зберігає некомпільовані ресурси, включаючи XML-файли розмітки інтерфейсу, зображення, рядкові константи та стилі оформлення.

6. Що таке графічна реалізація Activity?

Графічна реалізація Activity – це XML-файл розмітки (Layout), який декларативно описує структуру користувачького інтерфейсу. У цьому файлі визначаються візуальні елементи (View) та контейнери (ViewGroup), їх розташування, розміри та властивості, які потім завантажуються та відображаються класом Activity під час виконання програми.

Лістинг.

MainActivity.kt

```
1 package ua.kpi.lab1
2
3 import android.content.Intent
4 import android.os.Bundle
5 import android.widget.Button
6 import androidx.activity.enableEdgeToEdge
7 import androidx.appcompat.app.AppCompatActivity
8 import androidx.core.view.ViewCompat
9 import androidx.core.view.WindowInsetsCompat
10
11 class MainActivity : AppCompatActivity() {
12     override fun onCreate(savedInstanceState: Bundle?) {
13         super.onCreate(savedInstanceState)
14         enableEdgeToEdge()
15         setContentView(R.layout.activity_main)
16
17         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
18             val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
19             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
20             insets
21         }
22
23         val surname = "Ковалев"
24         val btn1 = findViewById<Button>(R.id.btn1)
25         btn1.setOnClickListener {
26             val intent = Intent(this, SecondActivity::class.java)
27
28             intent.putExtra("SURNAME_KEY", surname)
29
30             startActivity(intent)
31         }
32     }
33 }
```

activity_main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:id="@+id/main"
7     android:layout_width="match_parent"
8     android:layout_height="match_parent"
9     tools:context=".MainActivity">
10
11     <TextView
12         android:id="@+id/myTextView"
13         android:layout_width="wrap_content"
14         android:layout_height="wrap_content"
15         android:layout_marginBottom="20dp"
16         android:text="Activity 1"
17         android:textSize="24sp"
18
19         app:layout_constraintBottom_toTopOf="@+id	btn1"
20         app:layout_constraintEnd_toEndOf="parent"
21         app:layout_constraintStart_toStartOf="parent"
```

```

21    app:layout_constraintTop_toTopOf="parent"
22    app:layout_constraintVertical_chainStyle="packed" />
23
24    <Button
25        android:id="@+id	btn1"
26        android:layout_width="wrap_content"
27        android:layout_height="wrap_content"
28        android:text="Передати прізвище"
29
30        app:layout_constraintBottom_toBottomOf="parent"
31        app:layout_constraintEnd_toEndOf="parent"
32        app:layout_constraintStart_toStartOf="parent"
33        app:layout_constraintTop_toBottomOf="@+id/myTextView" />
34
35    </androidx.constraintlayout.widget.ConstraintLayout>
```

SecondActivity.kt

```

1 package ua.kpi.lab1
2
3 import android.os.Bundle
4 import android.widget.TextView
5 import androidx.activity.enableEdgeToEdge
6 import androidx.appcompat.app.AppCompatActivity
7 import androidx.core.view.ViewCompat
8 import androidx.core.view.WindowInsetsCompat
9
10 class SecondActivity : AppCompatActivity() {
11     override fun onCreate(savedInstanceState: Bundle?) {
12         super.onCreate(savedInstanceState)
13         enableEdgeToEdge()
14         setContentView(R.layout.activity_second)
15         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
16             val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
17             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
18             insets
19         }
20
21         val tvResult = findViewById<TextView>(R.id.tvResult)
22         val surname = intent.getStringExtra("SURNAME_KEY")
23         tvResult.text = "Параметр: $surname"
24     }
25 }
```

activity_second.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:id="@+id/main"
7     android:layout_width="match_parent"
8     android:layout_height="match_parent"
9     tools:context=".SecondActivity">
10
11     <TextView
12         android:id="@+id/secondActivityTV"
13         android:layout_width="wrap_content"
14         android:layout_height="wrap_content"
```

```
14     android:layout_marginBottom="20dp"
15     android:text="Activity 2"
16     android:textSize="24sp"
17
18     app:layout_constraintBottom_toTopOf="@+id/tvResult"
19     app:layout_constraintEnd_toEndOf="parent"
20     app:layout_constraintStart_toStartOf="parent"
21     app:layout_constraintTop_toTopOf="parent"
22     app:layout_constraintVertical_chainStyle="packed" />
23
24     <TextView
25         android:id="@+id/tvResult"
26         android:layout_width="wrap_content"
27         android:layout_height="wrap_content"
28         android:text="Чекаємо даних..."
29         android:textSize="24sp"
30         app:layout_constraintBottom_toBottomOf="parent"
31         app:layout_constraintEnd_toEndOf="parent"
32         app:layout_constraintStart_toStartOf="parent"
33         app:layout_constraintTop_toBottomOf="@+id/secondActivityTV" />
34
35     </androidx.constraintlayout.widget.ConstraintLayout>
```