Національний технічний університет України «КПІ ім. Ігоря Сікорського»

Факультет Інформатики та Обчислювальної Техніки

Кафедра інформаційних систем та технологій

Лабораторна робота №1

з дисципліни «Розробка мобільних застосувань під Android»

на тему

«ДОСЛІДЖЕННЯ РОБОТИ З ЕЛЕМЕНТАМИ КЕРУВАННЯ»

Виконав:

студент групи ІС-21

Качалов А. М.

Викладач:

Орленко С. П.

Київ - 2025

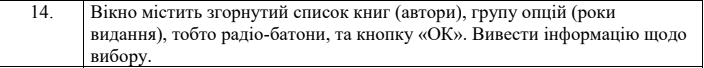
**Мета роботи**: дослідити створення простого застосунку під платформу Андроїд

та набути практичні навички з використання елементів керування інтерфейсу, мов

програмування Java чи Kotlin.

**ЗАВДАННЯ**

Написати програму під платформу Андроїд, яка має інтерфейс для введення або/та вибору даних згідно варіанту (таблиця) і відображає результат взаємодії з цим інтерфейсом у деяке текстове поле цього інтерфейсу. Передбачити наступне: якщо не всі дані введені або обрані, а користувач натискає кнопку для отримання результату, то відобразити вікно, що спливає, з повідомленням завершити введення всіх даних. Індивідуальне завдання, згідно із варіантом:



**Хід роботи**

1. Для виконання роботи, встановимо програму Android Studio, в якій будемо працювати зі створенням необхідного програмного забезпечення.

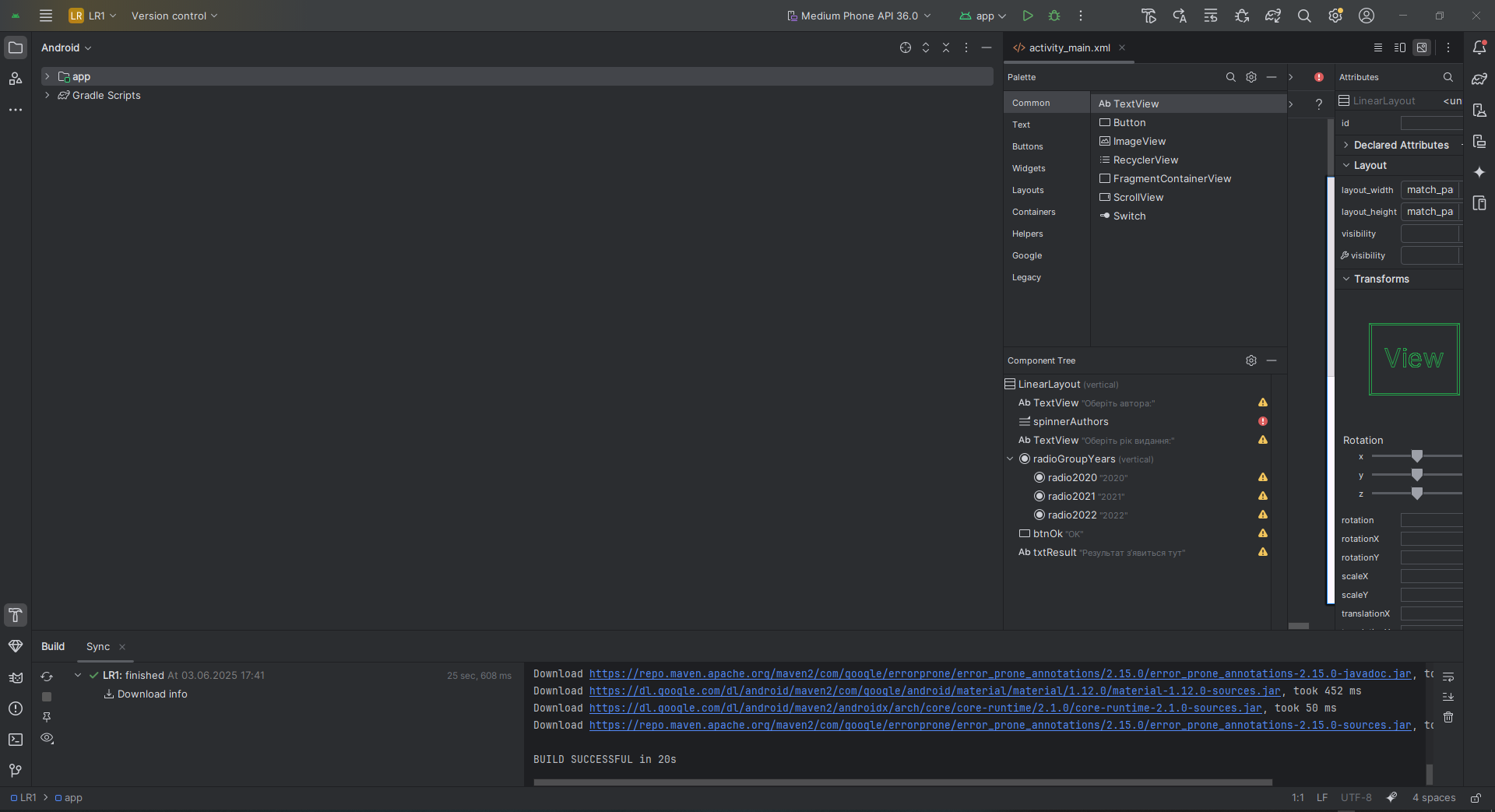


Рис. 1 – Загальний вигляд середовища Android Studio

2. Графічна складова проєкту представлена у вигляді файлу activity\_main.xml, в якому прописані необхідні елементи із завдання: список, радіо-батони, кнопка.

Для початку створюється группа об’єктів Linear Layout, в якій знаходяться всі елементи управління, такі як випадаючий список Spinner, селектори вибору Radio buttons, кнопка Button та текстові блоки TextView.

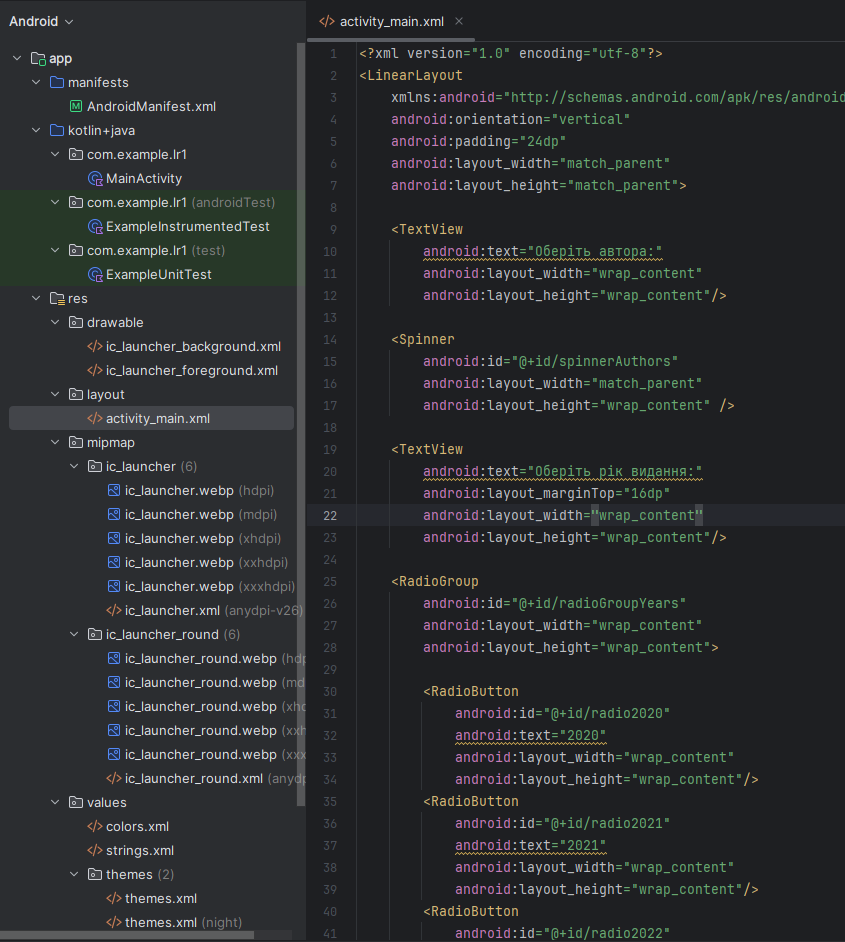


Рис. 2 – Скріншот activity\_main.xml

Після компіляції отриманого програмного коду, в середовищі Android Studio можна ознайомитись із отриманим графічним інтерфейсом створеної програми. В інтерфейсі знаходяться необхідні Text view, Radio Group (група Radio Button’ів), Button (кнопка «Ок»). Ознайомитись виглядом отриманого інтерфейсу можна на рисунку 3.

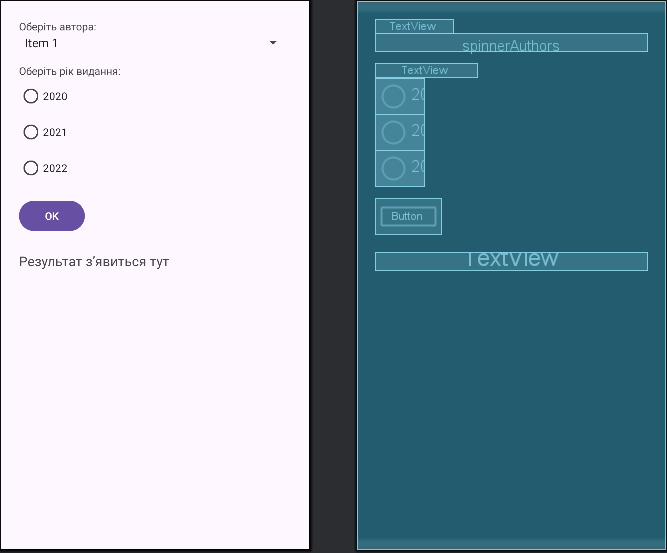


Рис. 3 – Загальний зовнішній вигляд ПЗ на екрані телефона зі схемою

3. Програмний код для роботи (а не тільки відображення) інтерфейсу знаходиться в файлі MainActivity. В ньому йде оголошення змінних для елементів UI, метод при створенні активності, підключення елементів до змінних, створення масиву для випадаючого списку, перетворення масиву у зрозумілий для випадаючого списку формат, встановлення даних у випадаючий список, обробник подій для натискання кнопки, зчитування вибору автора та року з відповідних полів, перевірка на заповнення полів, виведення повідомлення у разі помилки, зчитування тексту обраної користувачем кнопки, створення формату для виведеного тексту та виведення результату в текстове поле.

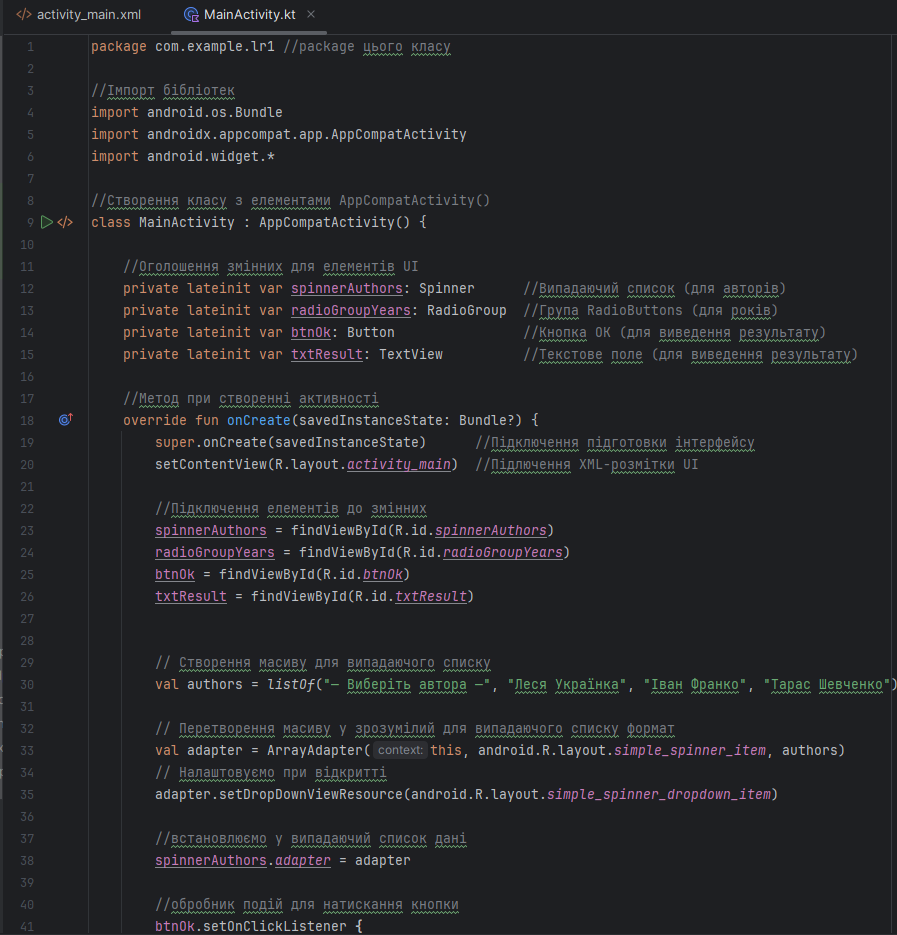
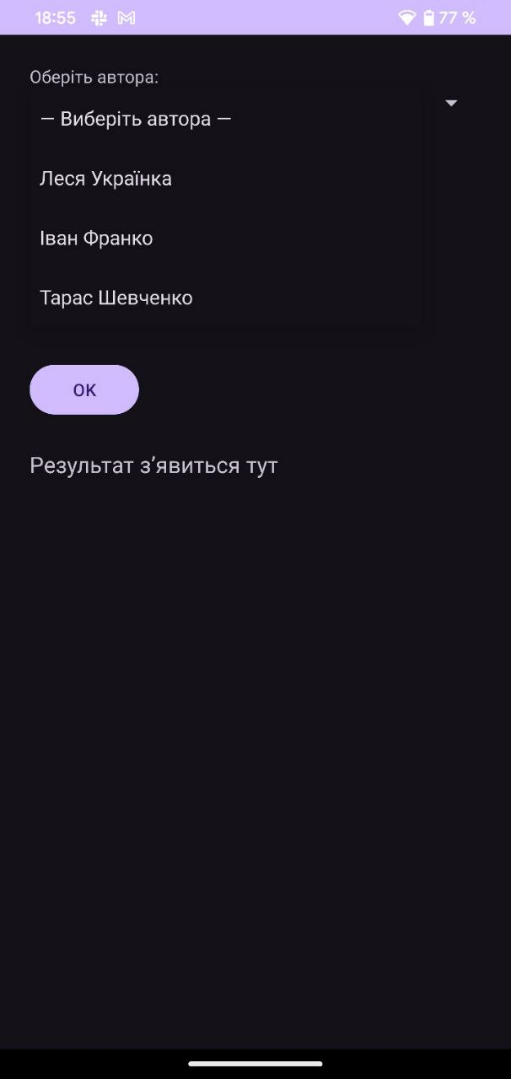
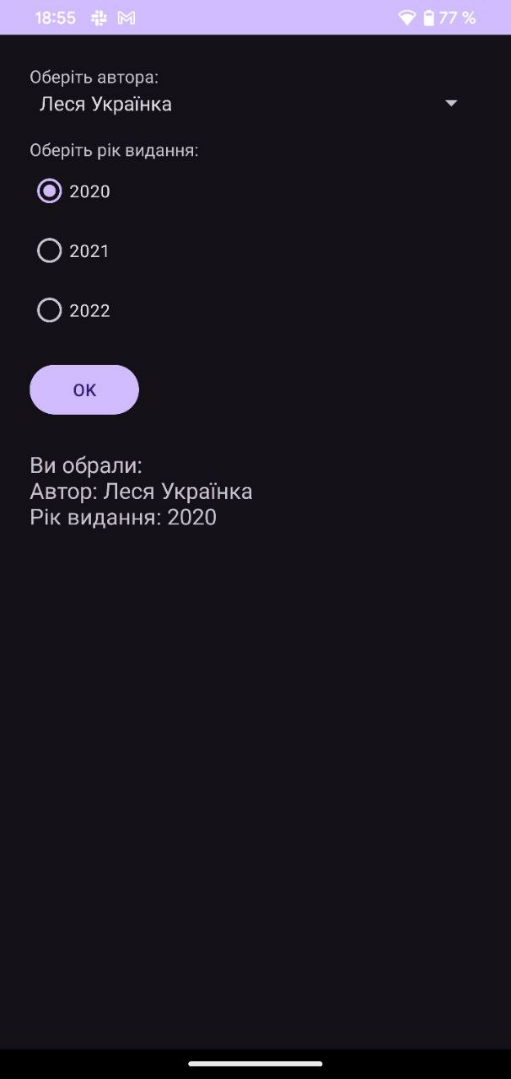
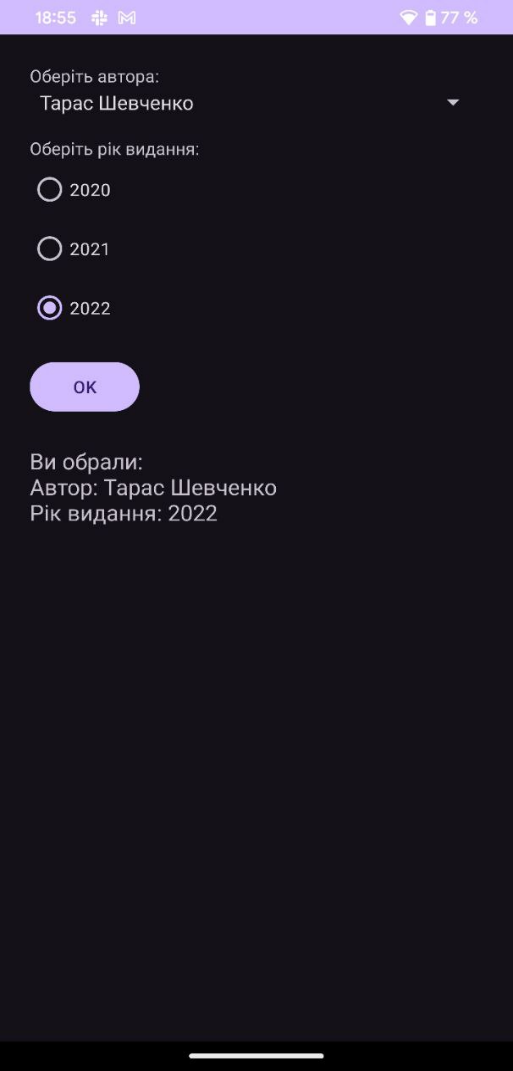


Рис. 4 – скріншот MainActivity.kt

4. Робота програмного продукту на телефоні

На скріншотах виконання програми видно всі елементи управління, а саме: випадаючий список, варіанти вибору в ньому, радіо-батони, текстові поля, кнопку та, звісно, виведення інформації. Випадаючий список із вмістом імен відомих українських письменників відкривається за натиском на нього, є можливість вибору кожного елемента, потім можна обрати рік видання, натиснувши на радіо-батон та, натиснувши на кнопку ОК, отримати свій вибір у текстовому полі під нею.

**Висновок**

Під час виконання даної лабораторної роботи, було досліджено роботу програми Android studio та створено простий застосунок під платформу Андроїд. В результаті, набуто практичні навички з використання елементів керування інтерфейсу, мови програмування Kotlin. Отриманий застосунок працює справно, в чому можна впевнитись на практиці, запустивши .apk файл, та виконує всі необхідні функції, які зазначені в умові виконання лабораторної роботи.

**Контрольні питання**

1. Архітектура застосунку під платформу Андроїд базується на багаторівневій моделі, де кожен рівень відповідає за окрему частину логіки та функціональності. Основними елементами архітектури є користувацький інтерфейс, прикладна логіка та система обміну даними з операційною системою. Застосунок запускається у власному процесі Android Runtime (ART), а його компоненти — такі як діяльності, служби, приймачі трансляцій і постачальники вмісту — функціонують у межах цього процесу. Для сучасної розробки рекомендується використовувати архітектуру MVVM (Model-View-ViewModel), яка дозволяє чітко розмежувати відповідальність між представленням, логікою і даними, забезпечуючи зручність тестування, масштабованість і підтримку реактивного підходу.

2. Загальний огляд компонентів застосунку під Андроїд включає чотири основні типи: діяльності (Activities), служби (Services), приймачі трансляцій (Broadcast Receivers) та постачальники вмісту (Content Providers). Діяльність відповідає за представлення інтерфейсу користувача та взаємодію з ним. Служби виконують фонові або тривалі завдання без взаємодії з інтерфейсом. Приймачі трансляцій дозволяють реагувати на системні події, такі як підключення до мережі або завершення завантаження. Постачальники вмісту забезпечують обмін даними між застосунками через уніфікований інтерфейс доступу до даних. Усі ці компоненти взаємодіють з системою Android через маніфест і можуть бути активовані як безпосередньо, так і за допомогою інтентів.

3. Життєвий цикл компоненту «Діяльність» визначає послідовність станів, через які проходить діяльність від створення до завершення. Основними методами цього циклу є onCreate(), де ініціалізуються компоненти інтерфейсу; onStart() та onResume(), які активують діяльність для взаємодії з користувачем; onPause() та onStop(), що викликаються при втраті фокусу або згортанні вікна; і onDestroy(), який звільняє ресурси перед завершенням. Крім того, метод onRestart() викликається, коли діяльність відновлюється після зупинки. Управління цим життєвим циклом дозволяє ефективно обробляти зміну конфігурації, збереження стану та оптимальне використання ресурсів пристрою.

4. Життєвий цикл компоненту «Служба» відрізняється від діяльності тим, що вона може функціонувати у фоновому режимі без інтерфейсу. Основними методами є onCreate(), який викликається при створенні служби, onStartCommand(), що обробляє кожен запит до служби, та onDestroy(), який викликається при її завершенні. Існує також варіант служби Bound Service, у якому служба зв’язується з компонентами за допомогою методу onBind() і працює лише під час активного зв’язку. Вибір між стандартною службою і зв’язаною залежить від потреб: для тривалого виконання у фоні підходить Started Service, а для тісної взаємодії з іншим компонентом — Bound Service.

5. Опис процесів платформи Андроїд пов’язаний із тим, що кожен застосунок працює у власному процесі Linux з унікальним ідентифікатором користувача (UID). Система Android автоматично створює процес для застосунку під час його запуску і закриває його при нестачі ресурсів або завершенні роботи застосунку. Компоненти застосунку, як-от діяльність або служба, запускаються у межах цього процесу. Android також підтримує запуск компонентів у фонових або окремих процесах, що можна вказати у маніфесті. Така модель забезпечує ізоляцію застосунків один від одного, підвищуючи безпеку та стабільність системи.

6. Компоненти застосунку активуються за допомогою системи інтентів — об'єктів, які описують дію, яку необхідно виконати, та дані, до яких вона застосовується. Діяльності зазвичай активуються явними інтентами, які точно вказують цільовий компонент. Служби можуть бути активовані як явними, так і неявними інтентами залежно від логіки застосунку. Приймачі трансляцій реагують на трансляції, які надсилаються системою або іншими застосунками. Постачальники вмісту активуються під час запиту на доступ до даних, які вони зберігають. Усі ці компоненти мають бути оголошені в маніфесті, що дозволяє системі Android знати про їхнє існування та правильно керувати їхнім життєвим циклом.

7. Призначення файлу маніфесту полягає в тому, щоб описати структуру, налаштування та дозволи застосунку. Це XML-файл з іменем AndroidManifest.xml, який є обов’язковим елементом кожного Android-проєкту. У ньому визначаються усі компоненти застосунку (діяльності, служби, приймачі трансляцій, постачальники вмісту), дозволи на доступ до інтернету, камери, геолокації та інші ресурси, а також мінімальна та цільова версія платформи. Крім того, у маніфесті можна вказати фільтри інтентів, дозволяючи компонентам реагувати на певні дії, наприклад відкриття файлу чи отримання системного повідомлення.

8. Поняття ресурсу в Android охоплює всі зовнішні дані, які не є частиною вихідного коду: це зображення, рядки, кольори, стилі, макети інтерфейсу, а також налаштування. Ресурси визначаються у спеціальних підкаталогах директорії res — наприклад, res/layout для XML-макетів, res/values для текстів, res/drawable для графічних ресурсів. Застосунок звертається до ресурсів через автоматично згенерований клас R, що дозволяє зручно отримувати доступ до них у коді. Такий підхід полегшує підтримку багатомовності, масштабованість інтерфейсу для різних розмірів екранів і забезпечує розділення логіки програми та її візуального представлення.