Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота № 1
з дисципліни
Розробка мобільних застосувань під Android

Виконав:

студент групи ІА-34:
Бородай А.С.

Перевірив:

ст. викладач
Орленко С.П.

Київ 2026

Тема: Дослідження роботи з елементами керування.

Мета: дослідити створення простого застосунку під платформу Андроїд та набути практичні навички з використання елементів керування інтерфейсу, мов програмування Java чи Kotlin.

Варіант: 5.

Завдання до роботи

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. Примітки: варіант завдання обирається за списком слухачів дисципліни (загальним, де 148 студентів, 18-й за списком обирає 1-й варіант, 19-й - друге завдання і т.д.). Також можна спробувати сформувати своє подібне завдання для якогось практичнішого застосування (але тоді ознайомтесь з завданнями на 2 та 3 роботи).

5.	Вікно замовлення квітів містить: текстове поле, дві групи опцій (колір, діапазон цін), тобто радіо-батонів та кнопку «ОК». Вивести інформацію щодо замовлення при натисканні на кнопку «ОК» у деяке текстове поле.

Хід роботи

GitGub: github.com/Borodai-Andrii-Stanislavovich/AndroidApplication

Створивши пустий проект у Android Studio, додаємо у єдину сторінку застосунку необхідні елементи (текстове поле, 2 групи радіо-батонів із радіо-батонами всередині, кнопку та інші додаткові елементи). До кожного елемента було вказано необхідні параметри за завданням (підказки, текст, положення та ін.).

Результат вигляду сторінки можна побачити у режимі дизайну xml файду activity_main (Рисунок 1.1).

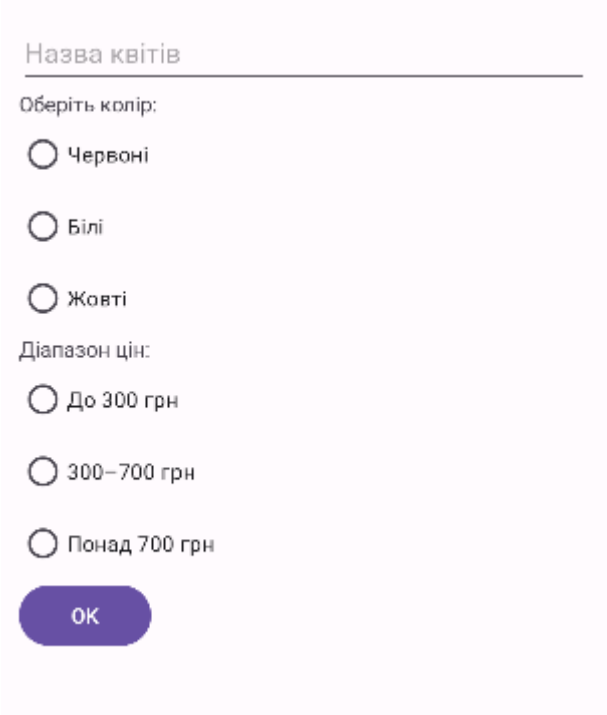


Рисунок 1.1 – Вигляд сторінки застосунку в Android Studio

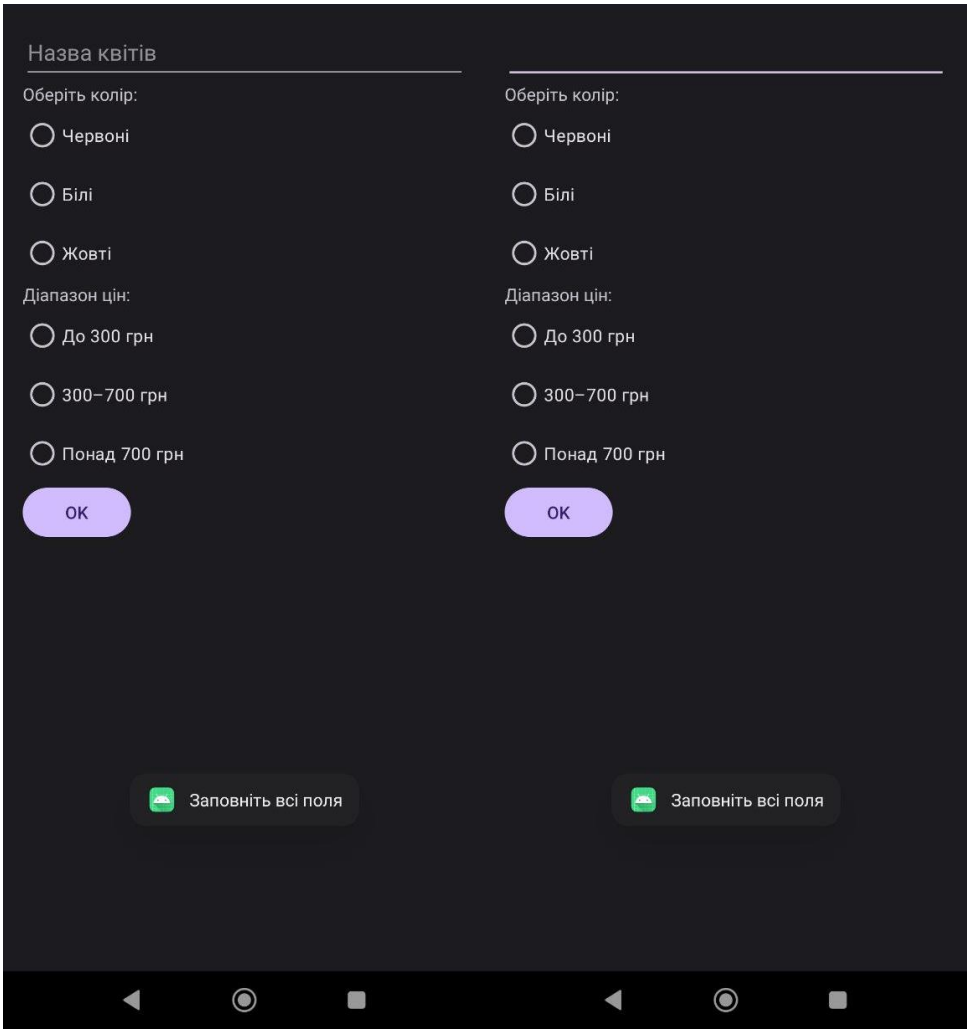
У файл MainActivity було додано логіку роботи із елементами сторінки застосунку.

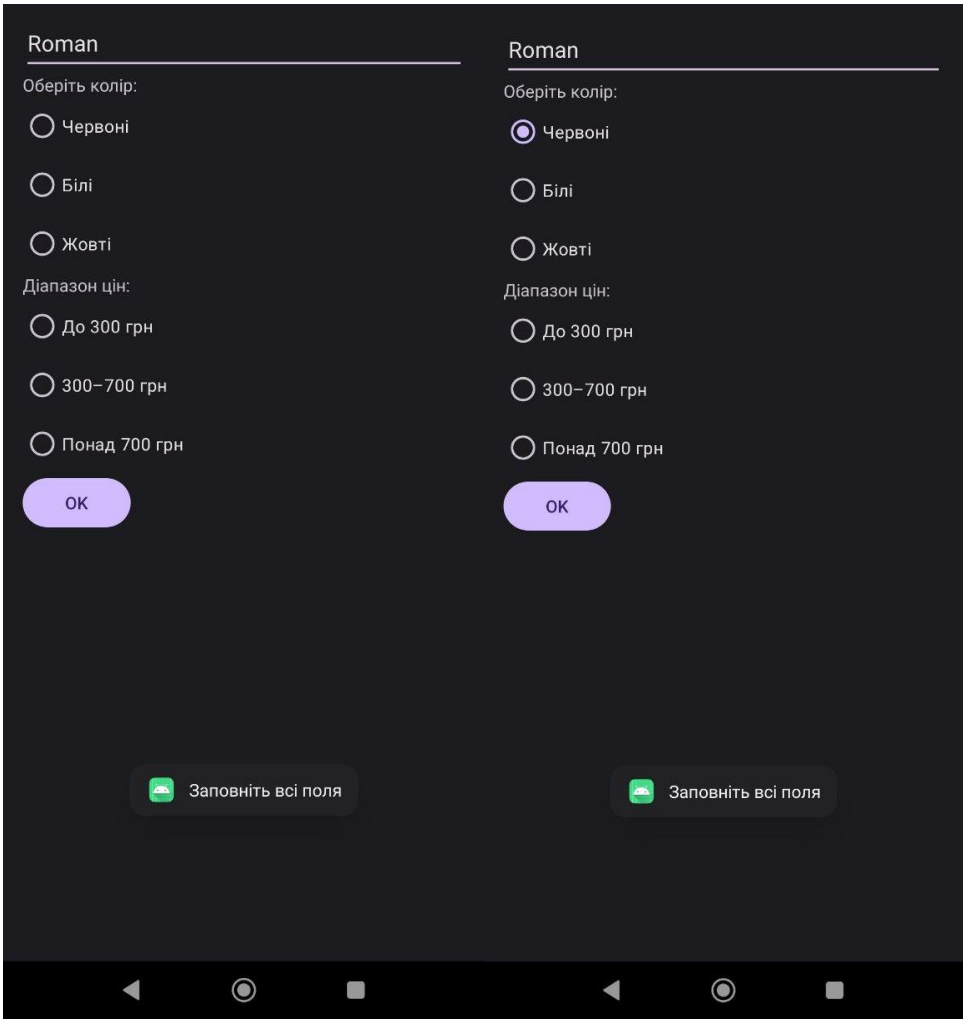
```
package com.example.myfirstapplication;

import android.os.Bundle;
import android.widget.*;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    EditText etFlowerName;
    RadioGroup rgColor, rgPrice;
    TextView tvResult;
    Button btnOk;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        etFlowerName = findViewById(R.id.etFlowerName);
        rgColor = findViewById(R.id.rgColor);
        rgPrice = findViewById(R.id.rgPrice);
        tvResult = findViewById(R.id.tvResult);
        btnOk = findViewById(R.id.btnOk);

        btnOk.setOnClickListener(v -> {
            String flower = etFlowerName.getText().toString().trim();

            if (flower.isEmpty()
                    || rgColor.getCheckedRadioButtonId() == -1
                    || rgPrice.getCheckedRadioButtonId() == -1) {
```

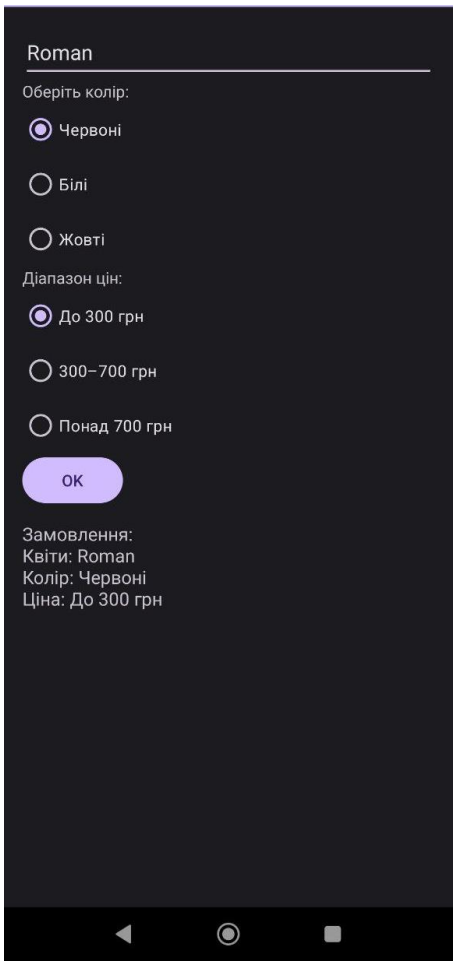
```
                Toast.makeText(this,
                        "Заповніть всі поля",
                        Toast.LENGTH_SHORT).show();
                return;
            }

            RadioButton colorBtn =
findViewById(rgColor.getCheckedRadioButtonId());
            RadioButton priceBtn =
findViewById(rgPrice.getCheckedRadioButtonId());

            String result = "Замовлення:\n"
                    + "Квіти: " + flower + "\n"
                    + "Колір: " + colorBtn.getText() + "\n"
                    + "Ціна: " + priceBtn.getText();

            tvResult.setText(result);
        });
    }
}
```

Запустимо застосунок через емулятор. Перевіряємо коректність роботи.

Якщо не заповнити текстове поле (або заповнити пробілами), не обрано обидва варіанти, то показується підказка. Інакше виводиться дані замовлення.




Roman
Оберіть колір:
Червоні
Білі
Жовті
Діапазон цін:
До 300 грн
300–700 грн
Понад 700 грн
OK
Заповніть всі поля
Roman
Оберіть колір:
Червоні
Білі
Жовті
Діапазон цін:
До 300 грн
300–700 грн
Понад 700 грн
OK
Заповніть всі поля


Roman
Оберіть колір:
Червоні
Білі
Жовті
Діапазон цін:
До 300 грн
300–700 грн
Понад 700 грн
OK
Замовлення:
Квіти: Roman
Колір: Червоні
Ціна: До 300 грн

Висновки про результати досліджень

У ході виконання лабораторної роботи було розроблено простий Android-застосунок у середовищі Android Studio.

Реалізовано графічний інтерфейс із базовими елементами керування, забезпечено введення та вибір даних користувачем, а також перевірку коректності заповнення полів.

Отримано практичні навички роботи з розміткою інтерфейсу, обробкою подій та взаємодією між елементами інтерфейсу. Поставлену мету роботи досягнуто.

Контрольні питання

1. Архітектура застосунку під платформу Андроїд.

Додаток на Андроїд це спеціальний файл архів з розширенням apk, який являє собою набір компонентів. Програма може використовувати компоненти інших програм (з дозволу), але вона не містить код інших програм та посилання на них.

Є збирач сміття, який знищує об’єкти із якими немає взаємодії, або операційно пам’яті становиться замало. ОС може знищити процес, якщо він не показує користувачу ніякого графічного інтерфейсу.

Є стек користувача для запам’ятовування попередніх інтерфейсів. Обробка взаємодії між інтерфейсом (користувача) та логікою програми за шаблоном MVVM.

2. Загальний огляд компонентів застосунку під Андроїд.

Компонентами Андроїд застосунку є: діяльність, служба, приймач широкомовних намірів, контент провайдер.

Діяльність – Візуальний інтерфейс (вікно, екран) для дій, які користувач може зробити. Діяльностей може бути кілька у застосунку. Діяльність може використовувати додаткові вікна.

Служба – компонент, який виконується у фоновому режимі протягом невизначеного часу. Програми можуть звертатися до служби або запускати, а також зупиняти запущені. Коли є підключення до служби, то звернення до її функцій здійснюється через інтерфейс, наданий цією службою.

Контент провайдер – компонент, що забезпечує доступ до даних програми з інших програм (тобто можна конфігурувати власні контент-провайдери, щоб дозволити доступ до своїх даних з інших програм; або використовувати контент провайдери інших програм для доступу до їх сховищ даних).

Приймач широкомовних намірів – компонент, що використовується для отримання зовнішніх подій і реакції на них (тобто програма може мати будь-яке число приймачів, щоб відповісти на будь-які повідомлення, як вважає важливими).

Наприклад, про зміни в стані мережевого підключення або рівень заряду батареї.

Приймачі широкомовних намірів не мають користувацького інтерфейсу, однак можуть запускати діяльність у відповідь на отримане повідомлення або показати повідомлення для інформування користувача.

Компоненти: діяльність, служба та приймач широкомовних намірів - активуються через асинхронні повідомлення – наміри. Наміри пов'язують різні компоненти один з одним у реальному часі (безвідносно того, чи ці компоненти є частиною однієї програми або різних). Контент-провайдер активується не наміром, а запитом від класу ContentResolver

3. Життєвий цикл компоненту «Діяльність».

→ Діяльність створюється методом onCreate()

→ Створена діяльність запускається з onStart() (*стає видимою)

→ Із діяльністю можна працювати після onResume() – знаходиться на передньому плані та має фокус для взаємодії з користувачем.

При onStart() та onResume() діяльність є видимою.

Діяльність можна призупинити onPause() (*після виклику onResume()) – діяльність втратила фокус, видна користувачу (або прозора), але може бути знищена системою у разі нестачі пам’яті.

Діяльність можна зупинити onStop() – діяльність повністю перекрита іншою діяльністю (буде знищення, якщо знадобиться память)

Діяльність знищується при виклику onDestroy().

4. Життєвий цикл компоненту «Служба».

2 види служб: started та bound.

→ Служба створюється методом onCreate()

→ Створена служба, якщо вона “ started ” запускається з onStartCommand() та використовується іншими службами. Служба може або зупинити сама себе, або якщо інший компонент її не зупинить stopService().

→ Створена служба, якщо вона “ bound ” запускається з onBind() та працює поки є клієнти. Якщо клієнтів немає то йде onUnbind() та onDestroy().

→ Із діяльністю можна працювати після onResume() – знаходиться на передньому плані та має фокус для взаємодії з користувачем.

Служба знищується при виклику onDestroy().

5. Опис процесів платформи Андроїд.

Система запускає процес за необхідності одного із компонентів. Усі компоненти ініціалізуються в основному потоці.

Потік може завершити своє виконання системою у разі нестачі пам’яті, або якщо пам’яті важливішим процесам не вистачає.

Перед знищенням проводиться оцінка діяльності процесу.

Пріоритети по критичності: активний процес, видимий процес, сервісний процес, фоновий процес, пустий процес.

Активний процес – це процес з яким користувач взаємодіє у даний момент. Вона виконує службу, пов’язану із діяльністю, з якою взаємодіє користувач або містити службу разом із якою виконує метод зворотного зв’язку або процес містить приймач широкомовних намірів.

Видимий процес - процес, що має діяльність, видиму кінцевому користувачеві вданий момент часу (діяльність втратила фокус введення, але ще видна користувачеві, або має службу, пов'язану на даний момент з діяльністю, яка перебуває на передньому плані (або частково перекрита).

Сервісний (службовий) процес - це процес, що містить, виконувану на даний момент службу, яка не відноситься до попередніх типів.

Фоновий процес - це процес, що містить діяльність, яку не видно користувачеві (є безліч фоновихпроцесів, робота яких завершується за принципом "останній запущений закривається останнім").

Порожній процес - це процес, який не містить жоднихактивних компонентів програми і використовується як «кеш» для зменшення витрат під час виклику компонента.

Особливості: Якщо в одному процесі виконуються кілька компонент, то система оцінює пріоритет процесу по компоненті з найвищим пріоритетом. Процес, який обслуговує інший процес, має пріоритет не нижче обслуговуваного процесу. Якщо для виконання фонової діяльності необхідний тривалий час, то її краще запустити як окрему

службу, а не породити в потоці з цією діяльністю (сервісний потік має більш високий пріоритет, ніж фоновий)

6. Яким чином активуються компоненти застосунку.

Оскільки система запускає кожну програму в окремомупроцесі з правами доступу до файлів, які обмежують доступдо них іншим програмам, програми не можутьбезпосередньо активувати компонент іншої програми.

Однак, система Android може. Тому, щоб використовуватикомпонент іншої програми, необхідно повідомити систему,що є Намір (Intent) запустити компонент якоїсь програми, і система запустить цей компонент.

Явні наміри - визначають адресата по імені, мають набір значень і використовуються в основному для повідомлень всередині однієї програми.

Неявні наміри - не визначають адресата і використовуютьсяпереважно для активації компонентів в іншому додатку. При відсутності адреси система Android переглядає

Фільтри-Намірів всіх програм і знаходить той компонент, фільтр-намірів якого найбільше підходить для його виконання.

Об'єкт Intent (намір) містить інформацію, що становить інтерес:

1) для компонента, який отримує намір та дані, які передаються цьому компоненту;

2) для системи Android – ім'я компонента, який повинен обробити намір і набір параметрів запуску цього компонента.

7. Призначення файлу маніфесту та його структура.

Він надає системі основну інформацію пропрограму:

- визначає ім'я пакета програми (унікальний ідентифікатордля програми);
- описує компоненти програми Activities, Services, BroadcastReceivers та Content Providers (визначає імена класів, що реалізують кожен із компонентів та оголошує можливості);
- містить список необхідних дозволів для звернення до захищених частин API та взаємодії з іншими програмами;
- оголошує дозволи, які сторонні додатки повинні мати для взаємодії з компонентами цієї програми;
- оголошує мінімальний рівень API Android, необхідний дляроботи додатку;

- перераховує зв'язані бібліотеки.

Для визначення компонентів використовуються:

<activity> для Activity (діяльності)

<service> для Service (служби)

<receiver> для Broadcast Reciever (приймача широкомовних повідомлень)

<provider> для Content Providers (постачальники даних)

Якщо компоненти не заявлені в маніфесті, то вони не помітні системі і, отже, ніколи не можуть бути запущені.

Крім того, Broadcast Reciever може створюватися динамічно в коді та реєструватися за допомогою виклику registerReceiver()

8. Поняття ресурсу та яким чином визначаються ресурси.

Використання ресурсів дає можливість змінювати деякі частини програми без модифікації вихідного коду, а також дозволяє оптимізувати програму для різних пристроїв (з різною мовою інтерфейсу або розміром екрану)

Типи ресурсів:

- Зображення;
- Шари GUI (XML файли);
- Оголошення меню (XML файли);
- Текстові рядки.

Для кожного ресурсу, включеного в додаток Android, визначається унікальний ідентифікатор (ціле число) у файлі R.java, яке можна використовувати для посилання на ресурс з коду або інших ресурсів визначених в XML.

Цей клас R генерується на основі заданих ресурсів і створюється під час компіляції програми (немає сенсу його редагувати вручну, тому що всі зміни будуть втрачені при повторній генерації).