Національний технічний університет України
«Київський політехнічний інститут ім. І. Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №1

з дисципліни «Розробка мобільних застосувань під Android»
на тему «Дослідження роботи з елементами керування»

Виконав:
Студент групи ІО-21
Крохун Б. В.
Залікова книжка №2109
Перевірив *Орленко С. П.*

Київ 2025 р.

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.

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.

Варіант 4:

Вікно містить згорнутий список та кнопки «ОК» і «Cancel». Список містить перелік мов програмування. Вивести обране значення зі списку при натисканні на кнопку «ОК» у деяке текстове поле та очистити при натисканні кнопки «Cancel».

Хід роботи:

Застосунок було розроблено за допомогою мови програмування Java.

Посилання на репозиторій: https://github.com/bohdan-k05/android-lab1

Вміст директорії проекту:

- MainActivity.java:

```
package com.example.lab1;

import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.TextView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
/*
 * Лабораторна робота №1
 * З дисципліни "Розробка мобільних застосувань під Android"
 * Виконав студент групи ІО-21 Крохун Богдан
 * Варіант: 4
 */
public class MainActivity extends AppCompatActivity {

    private String text;
    private String selected_language;
```

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

        Spinner spinner_languages = findViewById(R.id.spinner_languages);
        Button btn_ok = findViewById(R.id.btn_ok);
        Button btn_cancel = findViewById(R.id.btn_cancel);
        TextView text_select = findViewById(R.id.text_select);
        TextView text_language = findViewById(R.id.text_language);

        // Отримання списку мов з strings.xml та його запис у Spinner
        ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(
                this,
                R.array.language_list,
                android.R.layout.simple_spinner_item
        );

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spinner_languages.setAdapter(adapter);

        // Введення обраної мови програмування при натисканні на кнопку "OK"
        btn_ok.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                text = "Selected programming language";
                text_select.setText(text);

                selected_language =
spinner_languages.getSelectedItem().toString();
                text_language.setText(selected_language);
            }
        });

        // Очищення обраної мови програмування при натисканні на кнопку "Cancel"
        btn_cancel.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                text = "Select programming language:";
                text_select.setText(text);

                selected_language = "";
                text_language.setText(selected_language);
            }
        });
    }
}
```

- activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

```
    <Spinner
        android:id="@+id/spinner_languages"
        android:layout_width="330dp"
        android:layout_height="55dp"
        android:gravity="center"
        android:visibility="visible"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.44" />

    <Button
        android:id="@+id/btn_ok"
        style="@style/Widget.AppCompat.Button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="@string/ok"
        app:layout_constraintStart_toStartOf="@+id/spinner_languages"
        app:layout_constraintTop_toBottomOf="@+id/spinner_languages" />

    <Button
        android:id="@+id/btn_cancel"
        style="@style/Widget.AppCompat.Button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="4dp"
        android:text="@string/cancel"
        app:layout_constraintEnd_toEndOf="@+id/spinner_languages"
        app:layout_constraintTop_toBottomOf="@+id/spinner_languages" />

    <TextView
        android:id="@+id/text_select"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/select"
        app:layout_constraintBottom_toTopOf="@+id/spinner_languages"
        app:layout_constraintEnd_toEndOf="@+id/spinner_languages"
        app:layout_constraintStart_toStartOf="@+id/spinner_languages" />

    <TextView
        android:id="@+id/text_language"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:text="@string/empty"
        android:textSize="24sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toTopOf="@+id/text_select"
        app:layout_constraintEnd_toEndOf="@+id/text_select"
        app:layout_constraintStart_toStartOf="@+id/text_select" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

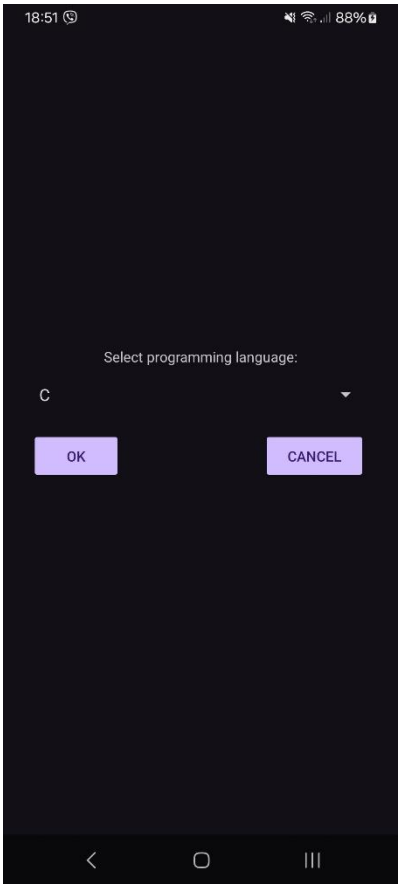
- strings.xml:

```
<resources>
    <string name="app_name">Lab 1</string>
    <string name="ok">OK</string>
    <string name="cancel">Cancel</string>
```

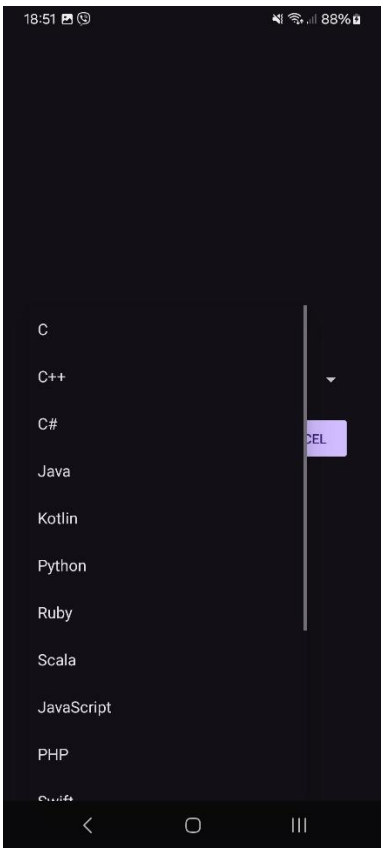
```
    <string name="select">Select programming language:</string>
    <string name="empty"/>
    <string-array name="language_list">
        <item>C</item>
        <item>C++</item>
        <item>C#</item>
        <item>Java</item>
        <item>Kotlin</item>
        <item>Python</item>
        <item>Ruby</item>
        <item>Scala</item>
        <item>JavaScript</item>
        <item>PHP</item>
        <item>Swift</item>
        <item>Go</item>
        <item>Rust</item>
        <item>SQL</item>
        <item>Pascal</item>
        <item>Lua</item>
    </string-array>
</resources>
```

Результати роботи програми:

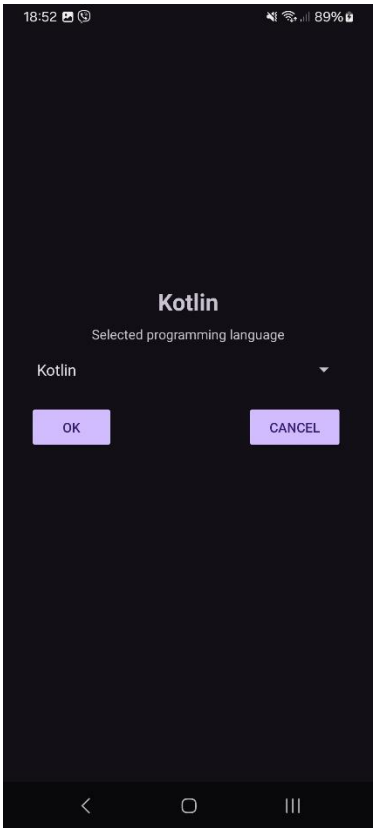
Початкове вікно:



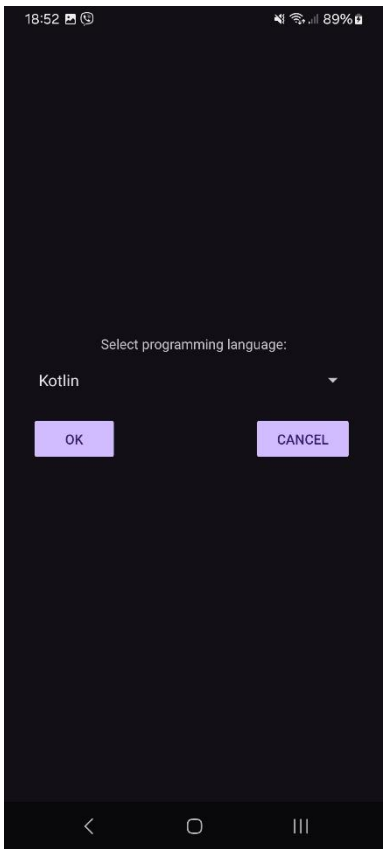
Для елементу вибору мови програмування було обрано Spinner. При натисканні на нього він відкриває список елементів, з яких необхідно обрати один:



Для прикладу оберемо Kotlin та натиснемо на кнопку “OK”:



Як можна побачити, відбулося виведення обраної мови програмування. Натиснемо кнопку “Cancel” та побачимо, що поле з обраною мовою програмування було очищено:



Висновок: На лабораторній роботі ми створили простий програмний застосунок для Android, що виводить обрані у елементі керування дані.

Відповіді на контрольні питання:

1. Архітектура застосунку під платформу Андроїд.

Архітектура застосунку під Android є framework-орієнтованою, тобто зводиться до розширення деяких класів або інтерфейсів, наданих фреймворком. Такий застосунок не може бути запущений без фреймворку або поза ним.

Вона має наступні особливості:

- Унікальна система управління пам’яттю - збирач сміття може знищити певні об’єкти, якщо ОС вирішить, що оперативної пам’яті мало, та ОС може знищити додаток, який в даний момент не показує користувачу ніякого інтерфейсу;
- Обробка приховування інтерфейсів користувача та існування кнопки «назад» на Android пристроях призводить до необхідності наявності стека інтерфейсів користувача, в якому поточний видимий інтерфейс поміщається на вершину, а всі інші зсуваються вниз;
- Обробка взаємодії між інтерфейсом користувача та його логікою відбувається за шаблоном Model – View – ViewModel.

2. Загальний огляд компонентів застосунку під Андроїд.

Android застосунок містить такі компоненти:

- Діяльність (Activity) – візуальний інтерфейс (вікно, екран) для однієї дії, яку може зробити користувач. Програма може складатися з лекількох діяльностей та кожна діяльність може використовувати додаткові вікна.
- Служба (Service) – не має візуального представлення, виконується у фоновому режимі протягом невизначеного часу (наприклад, програвач музики). Програми можуть запускати служби або до них підключатися за допомогою інтерфейсу, наданою цією службою.
- Приймач широкомовних намірів (Broadcast Receiver) – використовується для отримання зовнішніх подій і реакції на них (наприклад, зміну заряду батареї). Вони не мають візуального представлення, але можуть запускати Діяльність або показати повідомлення для інформування користувача.
- Контент-провайдер (Content Provider) – забезпечує доступ до даних програми з інших програм. Дані можуть зберігатися у файлах, базах даних SQLite та будь-якому іншому вигляді.

3. Життєвий цикл компоненту «Діяльність».

Діяльність може бути в одному з трьох станів:

- Відновлена (resumed) – знаходиться на передньому плані та має фокус для взаємодії з користувачем;
- Призупинена (paused) – втратила фокус, але все ще видна користувачу (поверху знаходиться інша діяльність, яка або прозора або закриває її частково). Призупинена діяльність повністю жива, але може бути знищена системою за умови нестачі пам’яті.
- Зупинена (stopped) – повністю перекрита іншою діяльністю та буде знищена у випадку нестачі пам’яті.

Діяльність містить наступні методи:

- onCreate() – викликається один раз при створенні діяльності.
- onStart() – викликається для переходу діяльності до взаємодії з користувачем (стає видимою).
- onResume() – передає весь фокус введення діяльності.
- onPause() – викликається, коли користувач вирішує перейти до роботи з новим вікном.
- onStop() – викликається, коли вікно стає невидимим для користувача.
- onRestart() – викликається після того, як діяльність була знову запущена користувачем.

- onDestroy() – викликається після завершення роботи.

4. Життєвий цикл компоненту «Служба».

Існує дві форми життєвого циклу служб: Started (запущена) та Bound (прив’язана).

- Запущена форма запускається методом startService() та працює до виклику методу stopSelf() собою або методу stopService() іншим компоненто програми.
- Прив’язана форма запускається методом bindService() та працює, поки хоча б один компонент залишився прив’язаним до неї.

5. Опис процесів платформи Андроїд.

- Як тільки один із компонентів буде необхідним, система запускає процес, що складається з одного потоку виконання.
- Усі компоненти ініціалізуються в основному потоці.
- Система може завершити виконання потоку у разі нестачі пам'яті або якщо пам'ять потрібна іншим, найважливішим процесам.
- При виборі процесу для знищення оцінюється його важливість з точки зору користувача (тобто насамперед завершиться процес із невидимими Діяльностями).

Типи процесів та їх пріоритети:

- Критичний пріоритет:
    - Активний процес – процес, з яким взаємодіє користувач, виконує службу, пов’язану з діяльністю, з якою взаємодіє користувач, містить службу та метод зворотнього виклику, визначений для даної служби, або тей, що містить приймач широкомовних намірів і виконує його метод зворотного виклику для прийому наміру.
- Високий пріоритет:
    - Видимий процес – має діяльність, видиму кінцевому користувачеві в даний момент часу (діяльність втратила фокус, але все ще видима) або має службу, пов’язану з діяльністю, що перебуває на передньому плані або частково перекрита.
    - Сервісний процес – процес, що містить виконувану в даний момент службу, що не відноситься до попередніх типів.
- Низький пріоритет:
    - Фоновий процес – містить діяльність, яку не видно користувачеві.

- Пустий процес – не містить жодних активних компонентів та використовується як кеш для зменшення витрат під час виклику компонента.

6. Яким чином активуються компоненти застосунку.

Оскільки система запускає кожну програму в окремому процесі з правами доступу до файлів, які обмежують доступ до них іншим програмам, програми не можуть безпосередньо активувати компонент іншої програми. Однак, система Android може. Тому, щоб використовувати компонент іншої програми, необхідно повідомити систему, що є Намір (Intent) запустити компонент якоїсь програми, і система запустить цей компонент.

- 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).
- Компонент - Контент-провайдер активується не Наміром, а запитом від класу ContentResolver.

Існують явні та неявні наміри. Явні визначають адресата по імені, мають набір значень і використовуються в основному для повідомлень всередині однієї програми. Неявні не визначають адресата і використовуються переважно для активації компонентів в іншому додатку.

7. Призначення файлу маніфесту та його структура.

Файл маніфесту AndroidManifest.xml надає системі основну інформацію про програму:

- визначає ім'я пакета програми (унікальний ідентифікатор для програми);
- описує компоненти програми Activities, Services, Broadcast Receivers 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 для роботи 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.