

**Міністерство освіти і науки України**

**Національний університет «Львівська політехніка»**

**Кафедра ЕОМ**



**Звіт**

**до лабораторної роботи №2**

**з дисципліни: «Програмні технології мобільних обчислень»**

**на тему: «Управління обчисленнями в ОС Android»**

**Варіант №8**

**Виконав студент групи КІ-44:**

**Козлюк Д.С.**

**Перевірив:**

**Бачинський Р.В.**

**Львів 2023**

**Мета:** Оволодіти навичками роботи з Android Activity та Activity stack.

**Завдання:**

1. Програма виводу етапів життєвого циклу Android Activity
2. Програма контролю етапів життєвого циклу Android Activity та запуску нової Activity

**Хід виконання**

**1. Схема життєвого циклу Android Activity**

Життєвий цикл Activity. Activity можна описати як вікно, що містить елементи графічного інтерфейсу, з якими може взаємодіяти користувач. Початок роботи activity тягне за собою виклики методів життєвого циклу, які визначені в класі цього activity.

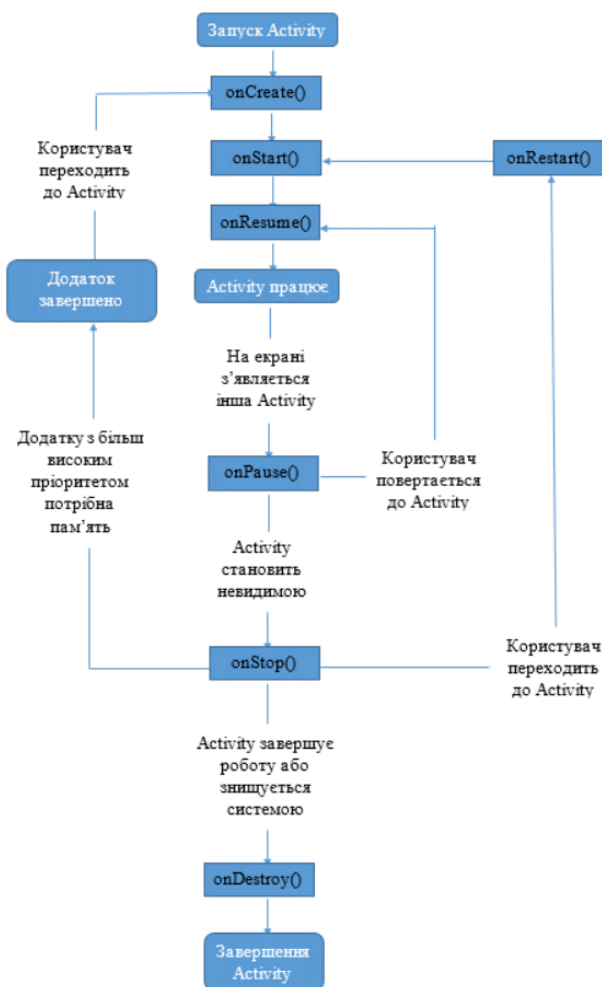


Рис.1. Життєвий цикл Activity

Система починає з виклику методу **onCreate** для створення activity. У цей метод слід помістити інструкції конструювання UI. Як тільки метод **onCreate** завершено, activity знаходиться у стані **Created**. Цей метод викликається один раз протягом життєвого циклу.

Далі операційна система викликає метод **onStart**. Коли цей метод викликано, activity становиться видимою. Після закінчення цього методу activity має стан **Started**. Цей метод може викликатися декілька разів під час життєвого циклу.

Метод **onResume** викликається перед тим, як activity починає взаємодіяти з користувачем. На цьому етапі activity знаходиться на вершині стека activity і в неї поступають дані, які вводить користувач. Стан activity після цього методу – **Resumed**. Метод **onResume** може викликатися декілька разів.

Отже, методи **onCreated**, **onStart** та **onResume** будуть визиватися послідовно у тому випадку, якщо не буде непередбачених виключень. У стані **Resumed** activity знаходиться у запущеному стані і залишається в ньому до тих пір, як якась подія не змінить його. Це може бути виклик, перехід пристрою у стан сну або запуск іншої activity.

**onPause** метод викликається тоді, коли activity змінює стан **Resumed**. Після закінчення методу activity переходить у стан **Paused**. **onPause** може викликатися багато разів протягом життєвого циклу.

Що далі відбувається після закінчення методу **onPause** залежить від того, чи буде далі activity повністю невидимою. Якщо так, викликається метод **onStop** і activity переходить до стану **Stopped**. Якщо activity знову викликається після **onPause**, система викликає метод **onResume** і activity знову переходить до стану **Resumed**.

Зі стану **Stopped** activity може бути відновлена, якщо користувач захочу повернутися до activity. В цьому випадку викликаються послідовно методи **onRestart** та **onStart**.

Нарешті, коли activity остаточно виведено з експлуатації, викликається метод **onDestroy**. Цей метод може визиватися один раз протягом життєвого циклу.

## 2. Лістинг тестової програми

### MainActivity.java

```
package com.example.lab2;

import android.content.Intent;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    StringBuilder builder = new StringBuilder();
    TextView textView1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        textView1 = findViewById(R.id.textView1);
        textView1.setText(builder.append("ЖИТТЄВИЙ ЦИКЛ:\n\n"));
        setTitle("Kozliuk_KI44_Lab2_2023");
        log("onCreate");
    }

    public void sendMessage(View view) {
        Intent intent = new Intent(this, ActivityTwo.class);
        startActivity(intent);
    }

    private void log(String text) {
        Log.d("LifeCycleTestActivity1", text);
        builder.append(text);
        builder.append('\n');
        textView1.setText(builder.toString());
    }

    @Override
    protected void onStart() {
        super.onStart();
        log("onStart");
    }

    @Override
    protected void onRestart() {
        super.onRestart();
        log("onRestart");
    }

    @Override
    protected void onStop() {
        super.onStop();
        log("onStop");
    }
}
```

```

@Override
protected void onDestroy() {
    super.onDestroy();
    log("onDestroy");
}

@Override
protected void onPause() {
    super.onPause();
    log("onPause");
    if (isFinishing()) {
        log("finishing");
    }
}

@Override
protected void onResume() {
    super.onResume();
    log("onResume");
}
}

```

## ActivityTwo.java

```

package com.example.lab2;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class ActivityTwo extends AppCompatActivity {

    StringBuilder builder = new StringBuilder();
    TextView textView2;

    private Button button1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_two);

        button1 = findViewById(R.id.button1);
        button1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });
        textView2 = findViewById(R.id.textView2);
        log("onCreate");
    }

    private void log(String text) {
        Log.d("LifeCycleTestActivity2", text);
        builder.append(text);
    }
}

```

```
        builder.append('\n');
        textView2.setText(builder.toString());
    }

    @Override
    protected void onStart() {
        super.onStart();
        log("onStart");
    }

    @Override
    protected void onRestart() {
        super.onRestart();
        log("onRestart");
    }

    @Override
    protected void onStop() {
        super.onStop();
        log("onStop");
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        log("onDestroy");
    }

    @Override
    protected void onPause() {
        super.onPause();
        log("onPause");
    }

    @Override
    protected void onResume() {
        super.onResume();
        log("Resume");
    }
}
```

## main\_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    android:gravity="center">

    <com.google.android.material.button.MaterialButton
        android:id="@+id/go"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/next"
        app:cornerRadius="4dp"
        app:icon="@drawable/ic_arrow_forward"
        app:iconPadding="8dp"
        app:iconGravity="textStart"
        app:backgroundTint="#FF5722"
        android:textColor="#FFFFFF"
        android:onClick="sendMessage"
        android:layout_marginBottom="16dp" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="18sp"
        android:textColor="#333333"
        android:layout_gravity="center"
        android:text="Your text goes here." />

</LinearLayout>
```

## activity\_two.xml

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    android:gravity="center">

    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/exit"
        android:backgroundTint="#FF5722"
        android:textColor="#FFFFFF"
        android:layout_marginBottom="16dp" />
```

```
<TextView
    android:id="@+id/textView2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:textColor="#333333"
    android:layout_gravity="center"
    android:text="Your text goes here." />

</LinearLayout>
```

### 3. Результати виконання тестової програми

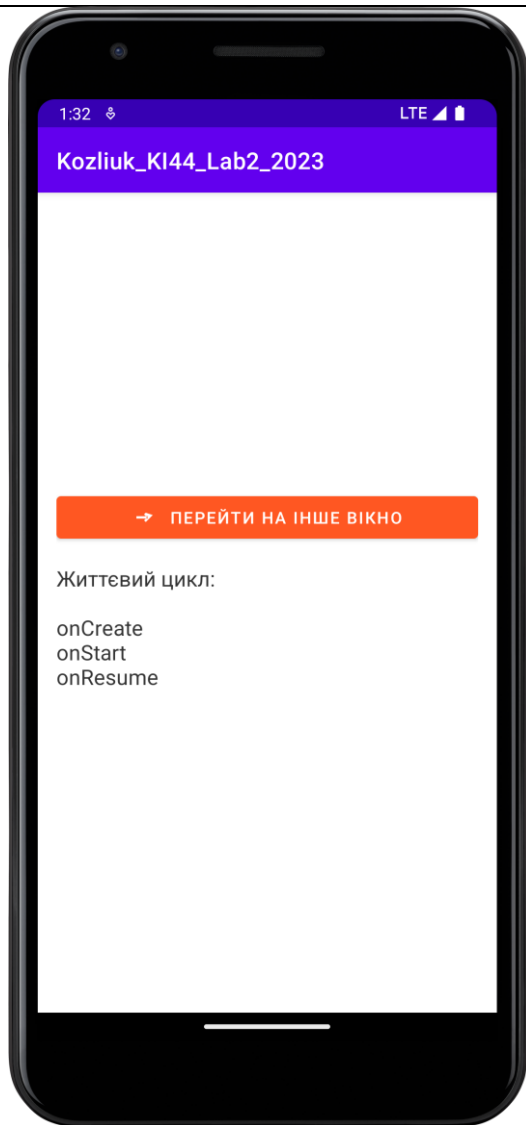


Рис.2. Життєвий цикл головного activity при створенні.

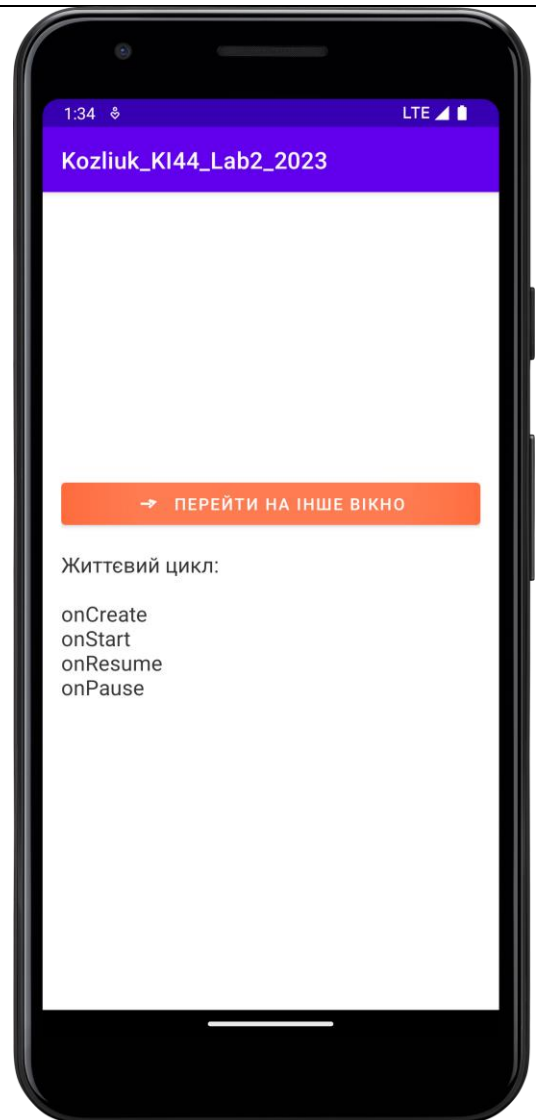


Рис.3. Життєвий цикл головного activity в момент переходу на інше activity



Як можна побачити на рис.3 що при переході на інше activity, головне activity переходить в стан onPause.



Рис.4. Життєвий цикл другого activity

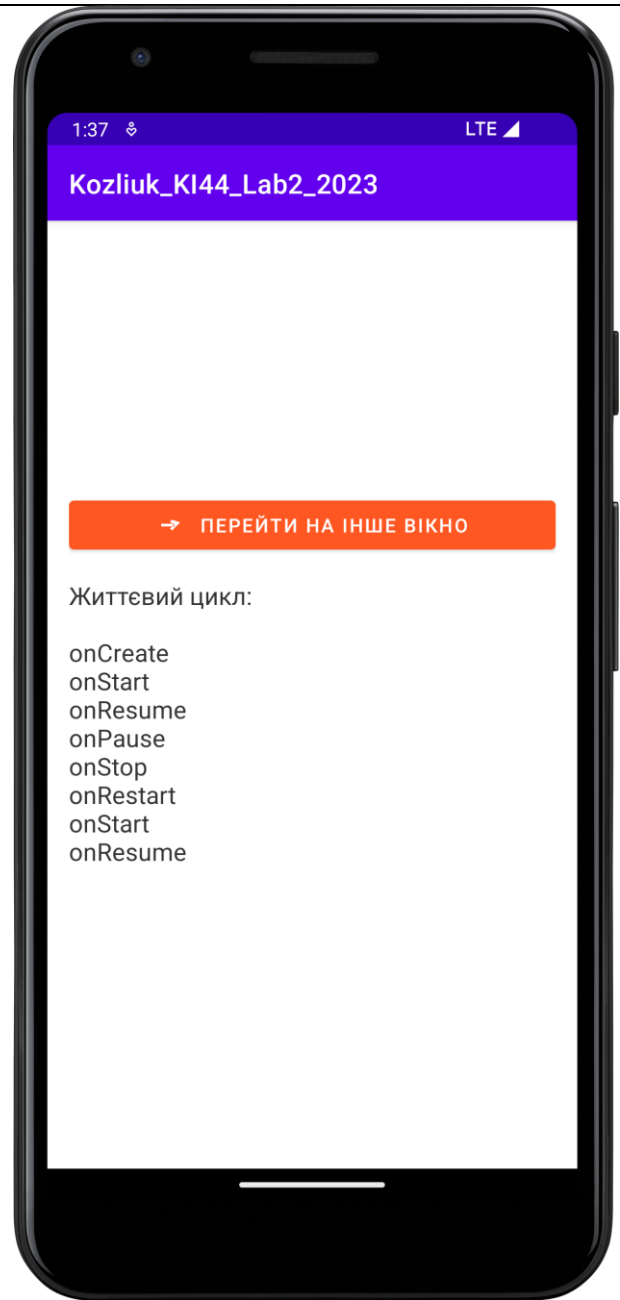


Рис.5. Життєвий цикл головного activity при переході із другого

Як можна побачити із рис.5 що коли друге activity повністю перекрило перше, то воно перейшло в onPause, а при переході на головне, життєвий цикл склався із 3-ох етапів: onRestart, onStart, onResume

```
2023-10-28 13:47:42.564 20675-20675 LifeCycleTestActivity2 com.example.lab2 D onCreate
2023-10-28 13:47:42.567 20675-20675 LifeCycleTestActivity2 com.example.lab2 D onStart
2023-10-28 13:47:42.569 20675-20675 LifeCycleTestActivity2 com.example.lab2 D onResume
2023-10-28 13:47:43.226 20675-20675 LifeCycleTestActivity1 com.example.lab2 D onStop
2023-10-28 13:47:44.194 20675-20675 LifeCycleTestActivity2 com.example.lab2 D onPause
2023-10-28 13:47:44.223 20675-20675 LifeCycleTestActivity1 com.example.lab2 D onRestart
2023-10-28 13:47:44.226 20675-20675 LifeCycleTestActivity1 com.example.lab2 D onStart
2023-10-28 13:47:44.230 20675-20675 LifeCycleTestActivity1 com.example.lab2 D onResume
2023-10-28 13:47:44.853 20675-20675 LifeCycleTestActivity2 com.example.lab2 D onStop
2023-10-28 13:47:44.860 20675-20675 LifeCycleTestActivity2 com.example.lab2 D onDestroy
2023-10-28 13:50:00.088 20675-20675 LifeCycleTestActivity1 com.example.lab2 D onPause
2023-10-28 13:50:00.170 20675-20675 LifeCycleTestActivity1 com.example.lab2 D onStop
2023-10-28 13:50:00.186 20675-20675 LifeCycleTestActivity1 com.example.lab2 D onDestroy
----- PROCESS ENDED (20675) for package com.example.lab2 -----
```

Рис.6. Повний життєвий цикл створеної програми

**Висновок:** Під час виконання даної лабораторної роботи, я дослідив роботу Android Activity та Activity stack. Створив тестову аплікацію для виводу етапів життєвого циклу та запуску нової Activity.