**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

**Факультет кібербезпеки, комп’ютерної та програмної інженерії**

Кафедра інженерії програмного забезпечення



Лабораторна робота №4

З дисципліни «**Системи для малих та моб платформ**»

на тему: «**Розмітки RelativeLayout і ConstraintLayout**»

Виконав студент:

групи ПІ-321

Іванюк Н. О.

Перевірив викладач:

Щерба В. В.

Київ 2021

**Теоретичні відомості**

У компонуваннях RelativeLayout та ConstraintLayout елементи інтерфейсу упорядковуються відносно контейнера (батьківського елемента), в якому вони знаходяться або відносно інших елементів цього контейнера.

ConstraintLayout (Android Studio за замовчуванням пропонує використовувати саме її) має додаткові можливості, особливо корисні при роботі у графічному режимі конструювання (вкладення Design). В Android Studio для цього контейнера є зручний візуальний редактор. Крім того ConstraintLayout, в більшості випадків, дає змогу уникнути зайвої вкладеності контейнерів, що позитивно впливає на продуктивність та читабельність коду. Цією розміткою можна, в певній мірі, замінити деякі розмітки, які з’явилися раніше.

У компонуванні ConstraintLayout існують різні типи прив'язок: по горизонталі, по вертикалі під певним кутом.

ConstraintLayout можна використовувати разом з ConstraintSet для одночасної анімації декількох елементів. ConstraintSet можна створити в коді або додати прямо в xml-файл.

До недоліків ConstraintLayout можна віднести:

 при великій кількості елементів і взаємозв'язків між ними складно зрозуміти, як вони всі взаємодіють, ніж у випадку ієрархічної вкладеності контейнерів;

 у конкретних випадках ця розмітка за продуктивністю може поступатися попереднім розміткам з вкладеними контейнерами, тому що додатково потрібен час на побудову дерева залежностей.

**Завдання**

1. Використовуючи розмітку RelativeLayout в XML-форматі, розташувати компоненти графічного інтерфейсу з вирівнюванням відносно батьківського елементу та інших елементів графічного інтерфейсу.

2. Використовуючи розмітку попереднього завдання за допомогою Java-коду видалити один з компонентів і вставити її в іншому місці екрану.

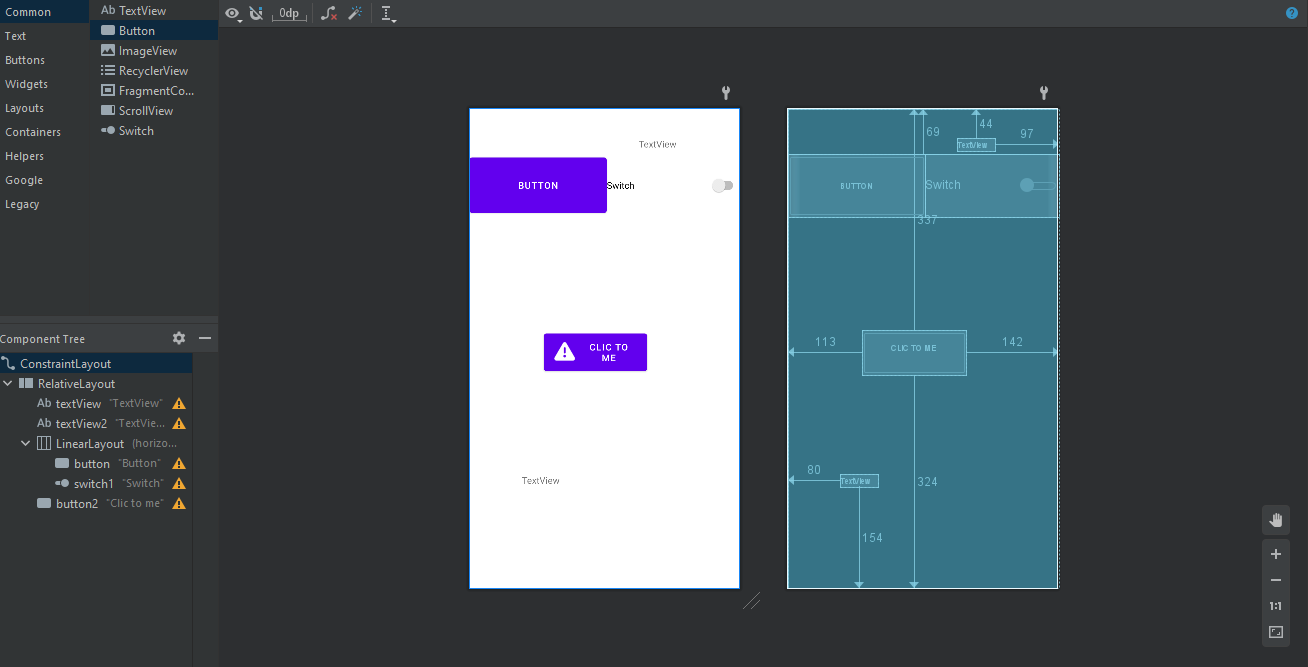
3. Використовуючи розмітку ConstraintLayout в режимі Design та XML-форматі, виконати завдання 1. Порівняти можливості RelativeLayout та ConstraintLayout.

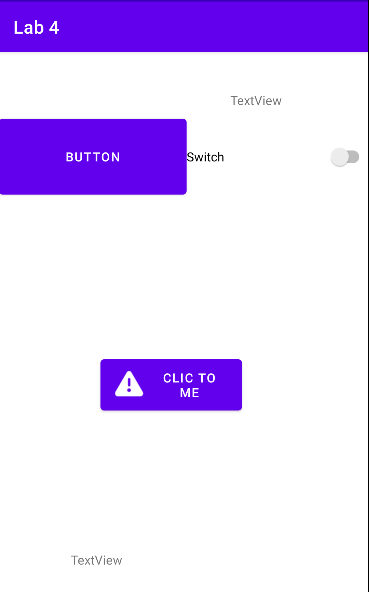
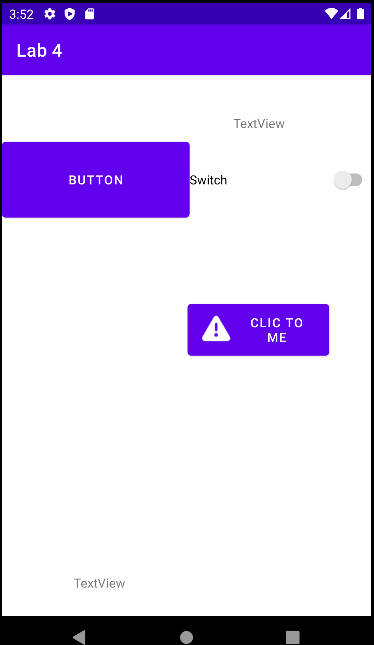
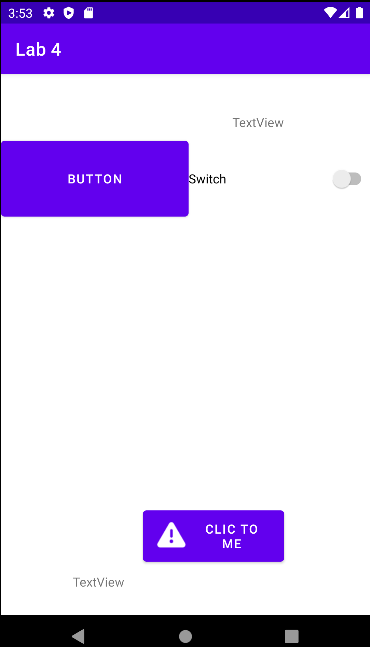
4. За допомогою ConstraintSet створити прив’язки, ланцюжки, барьєри, групи та використати constraintCircle. Використати ConstraintSet в коді або в xml-файлі.FrameLayout.

**Виконання**

ReiativeLayout (відносна розмітка) знаходиться в розділі Layouts і дозволяє дочірнім компонентів визначати свою позицію щодо батьківського компонента або щодо сусідніх дочірніх елементів (ідентифікатора елемента). У RelativeLayout дочірні елементи розташовані так, що якщо перший елемент розташований по центру екрана, інші елементи, вирівняні щодо першого елемента, будуть вирівняні відносно центру екрана.

Запуск програми:



**Код програми:**

<?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:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <RelativeLayout  
 android:layout\_width="413dp"  
 android:layout\_height="730dp"  
 tools:ignore="MissingConstraints"  
 tools:layout\_editor\_absoluteX="1dp"  
 tools:layout\_editor\_absoluteY="1dp">  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentBottom="true"  
 android:layout\_marginStart="80dp"  
 android:layout\_marginBottom="154dp"  
 android:text="TextView" />  
  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginTop="44dp"  
 android:layout\_marginEnd="97dp"  
 android:text="TextView" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="96dp"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginTop="69dp"  
 android:orientation="horizontal">  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="Button" />  
  
 <Switch  
 android:id="@+id/switch1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="96dp"  
 android:layout\_weight="1"  
 android:text="Switch" />  
 </LinearLayout>  
  
 <Button  
 android:id="@+id/button2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="68dp"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_alignParentBottom="true"  
 android:layout\_marginStart="158dp"  
 android:layout\_marginTop="481dp"  
 android:layout\_marginEnd="97dp"  
 android:layout\_marginBottom="180dp"  
 android:text="Clic to me"  
 app:icon="@android:drawable/ic\_dialog\_alert" />  
 </RelativeLayout>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

package com.example.lab4;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
  
import java.util.Random;  
  
public class MainActivity extends AppCompatActivity {  
 private Button button;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 button =findViewById(R.id.*button2*);  
  
 button.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Random random = new Random();  
 int x = random.nextInt(255);  
 int y = random.nextInt(661);  
 button.layout(x,  
 y,  
 255-x,  
 661-y);  
 }  
 });  
 }  
}

**Висновок**:

Виконавши лабораторну роботу , я дізнався про ConstraintLayout та RelativeLayoutм, навчився їх використовувати та програмувати їх у коді.