**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-324

Абідов Самір

Перевірив викладач:

Віктор Вікторович Щерба

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.

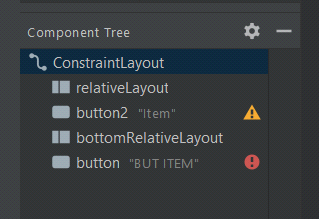
2. Використовуючи розмітку попереднього завдання за допомогою Java-коду видалити один з компонентів і вставити її в іншому місці екрану.

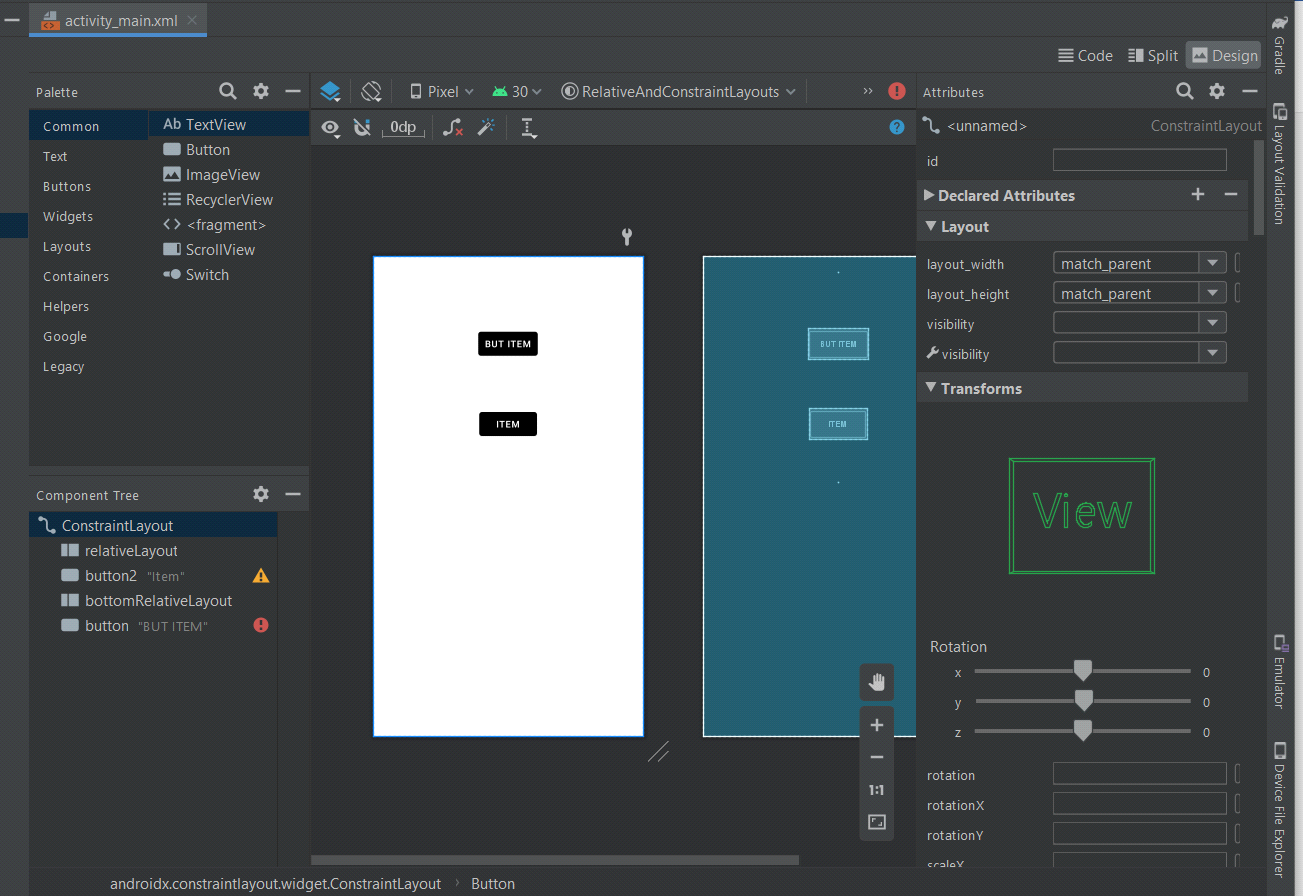
3. Використовуючи розмітку ConstraintLayout в режимі Design та XML-форматі, виконати завдання 1. Порівняти можливості RelativeLayout та ConstraintLayout.

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

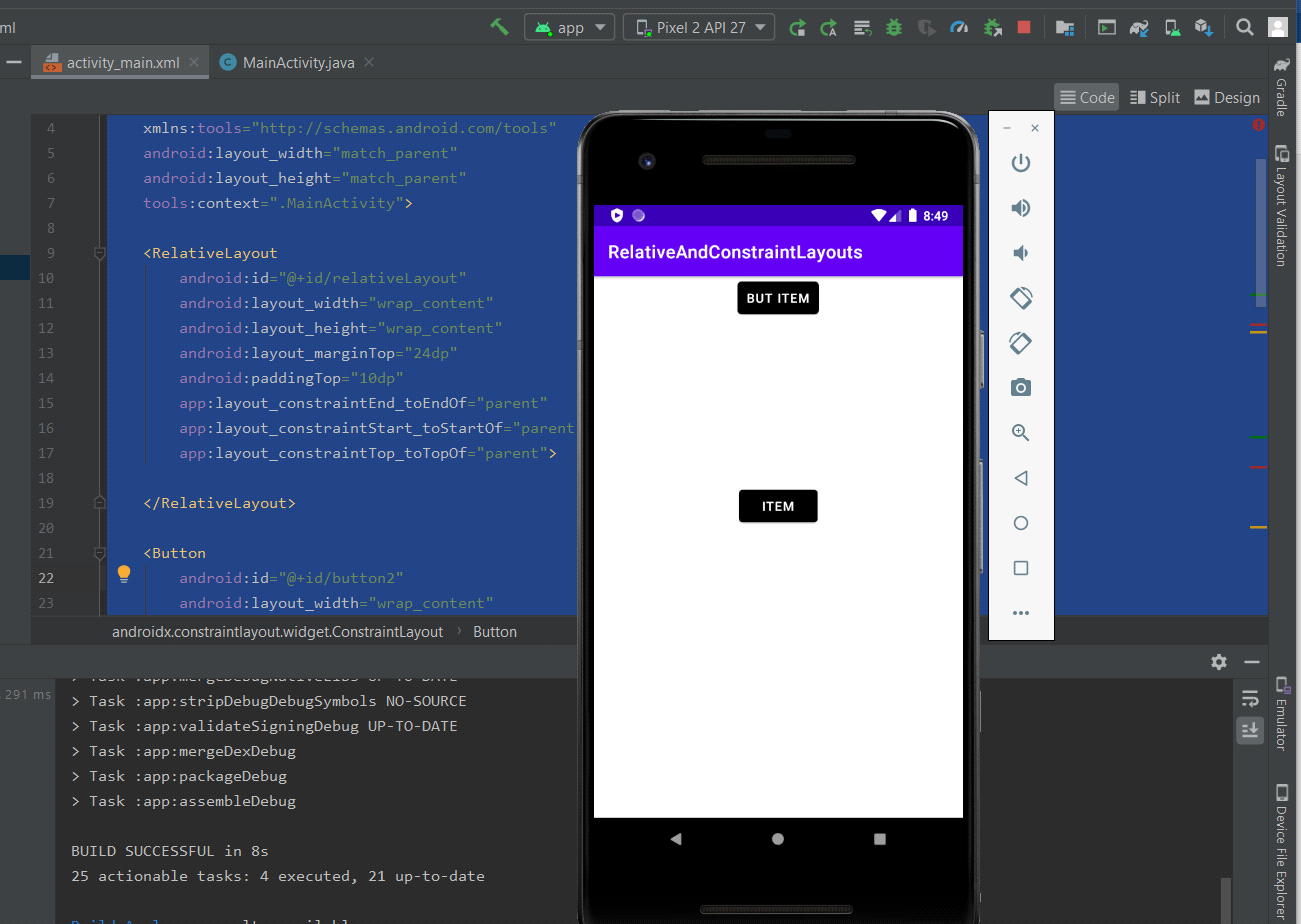
**Виконання**

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:id="@+id/relativeLayout"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="24dp"  
 android:paddingTop="10dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 </RelativeLayout>  
  
 <Button  
 android:id="@+id/button2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="208dp"  
 android:onClick="removeElement"  
 android:text="Item"  
 app:backgroundTint="@color/black"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/relativeLayout" />  
  
 <RelativeLayout  
 android:id="@+id/bottomRelativeLayout"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="64dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/button2">  
  
 </RelativeLayout>  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:paddingLeft="10dp"  
 android:paddingTop="5dp"  
 android:paddingRight="10dp"  
 android:paddingBottom="5dp"  
 android:text="BUT ITEM"  
 app:backgroundTint="@color/black"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 tools:layout\_editor\_absoluteY="110dp" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**Висновок**:

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