

Лабораторна робота №3. Робота з компонентами UI

Мета: створити проект з використанням віджетів.

Теоретичні відомості

Android SDK містить велику кількість UI (User Interface) компонентів (віджетів) (рис. 1).

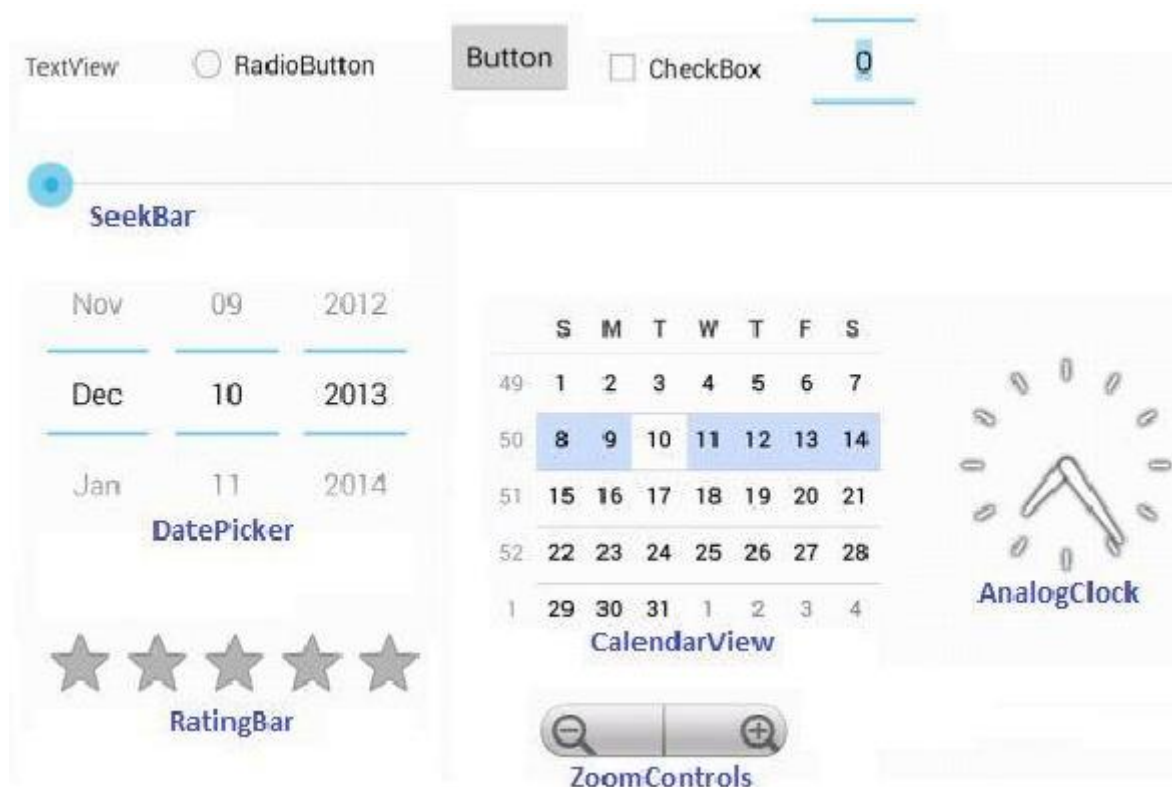


Рис. 1

Інструменти Android SDK (рис. 2) дозволяють автоматизувати створення графічного інтерфейсу.

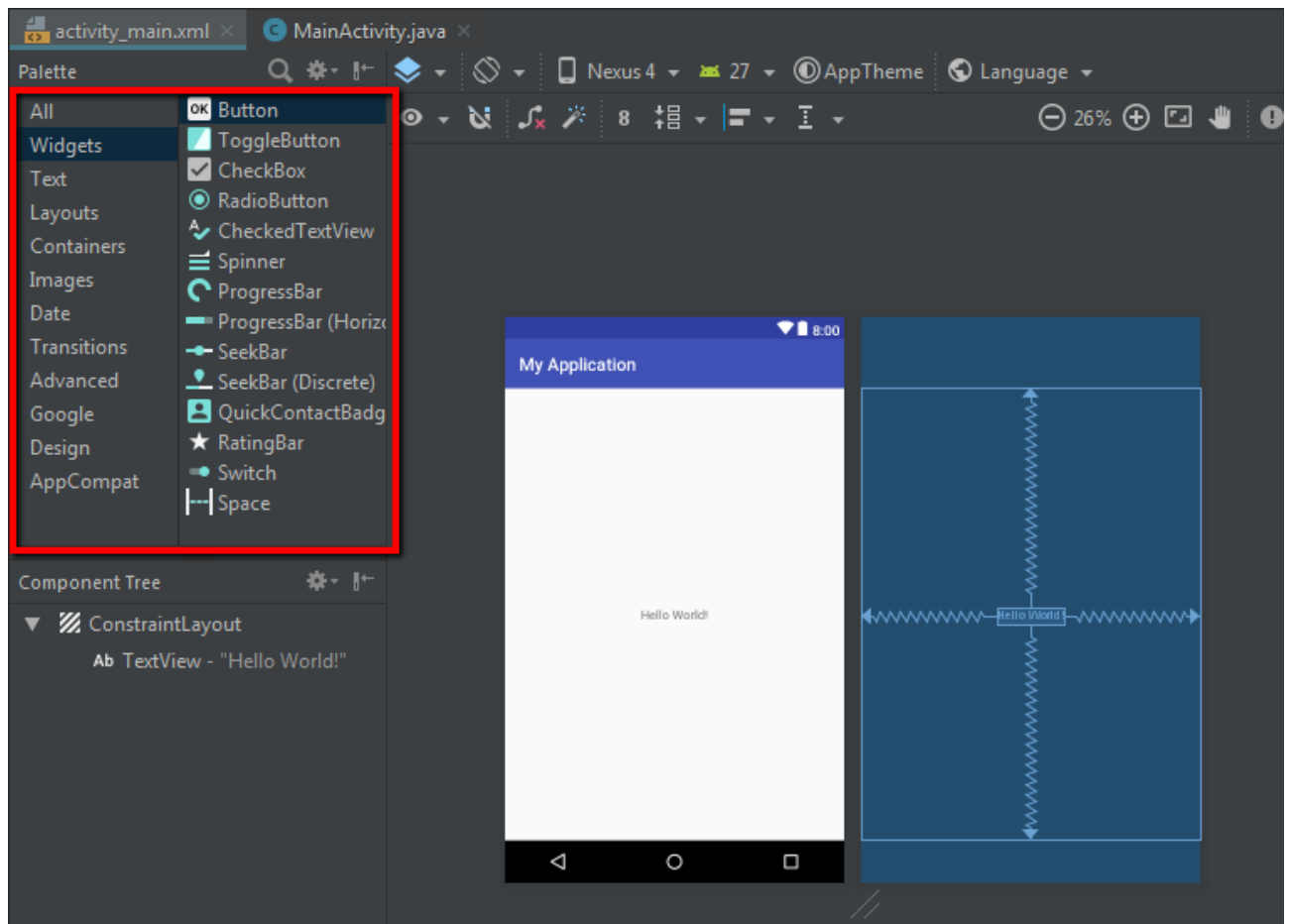


Рис. 2

Всі елементи графічного інтерфейсу в додатку Android створюються за допомогою об'єктів **View** (віджет) і **ViewGroup**. Найбільш поширений тип **ViewGroup** – це **layout**. В XML файлі кожного **layout** визначаються всі елементи **View**.

Зовнішній вигляд всіх віджетів можна визначати двома способами:

- У вікні **Attributes** при виділеному віджеті (рис. 3):
- У **XML** файлі відповідного **layout**.

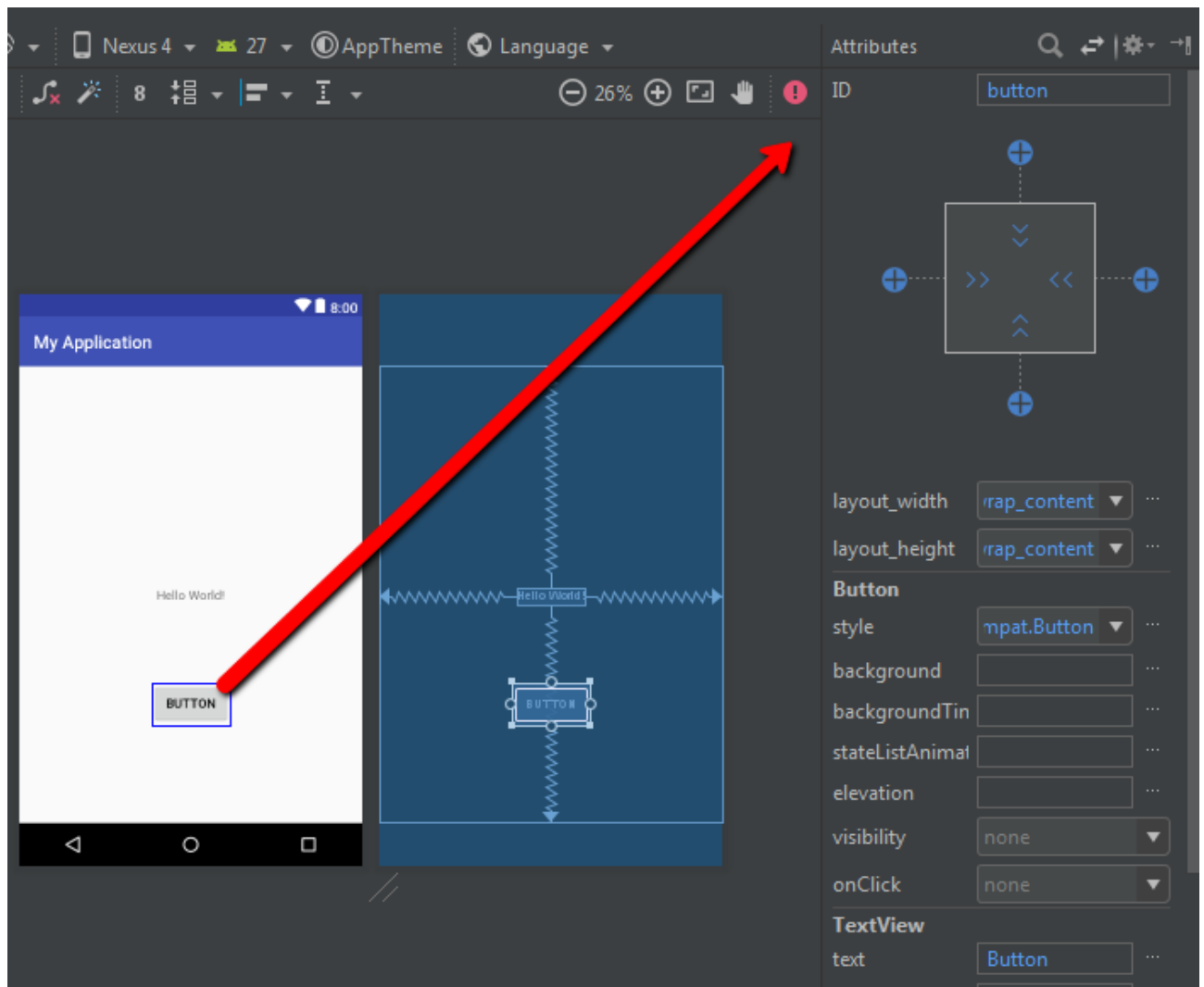


Рис. 3

При цьому, найбільш важливими властивостями є наступні:

- **width** (ширина);
- **height** (довжина);
- **weight** (вага);
- **gravity** (вирівнювання);
- **margin** (відступ);
- **padding** (відступ);
- **id** (ідентифікатор);
- **text**.

Вказані параметри можуть приймати наступні значення:

- **match_parent** (за розміром батьківського елемента);
- **wrap_content** (за розміром вмісту);
- **константа**.

Через константу позначаються параметри віджета у пікселях.

Є декілька понять, які пов'язані з відображенням даних на екрані пристрою.

Роздільна здатність екрана. Кількість точок по вертикалі та горизонталі, які екран здатний відобразити. Вимірюється у пікселях (px).

Густина пікселів – кількість пікселів в дюймі. Вимірюється в пікселях на дюйм (ppi).

Віртуальний піксель (density-independent pixel, dp) – відносне поняття, яке не залежить від фізичного розміру екрана. При визначення шрифтів використовується позначення sp (scaleable pixels).

На практиці, при завданні розмірів елементів екрана, зазвичай використовуються позначення у віртуальних пікселях (dp).

Наприклад, при завданні розміру кнопки 150x200 dpi для екранів з різною роздільною здатністю маємо приблизно однаковий розмір кнопки:

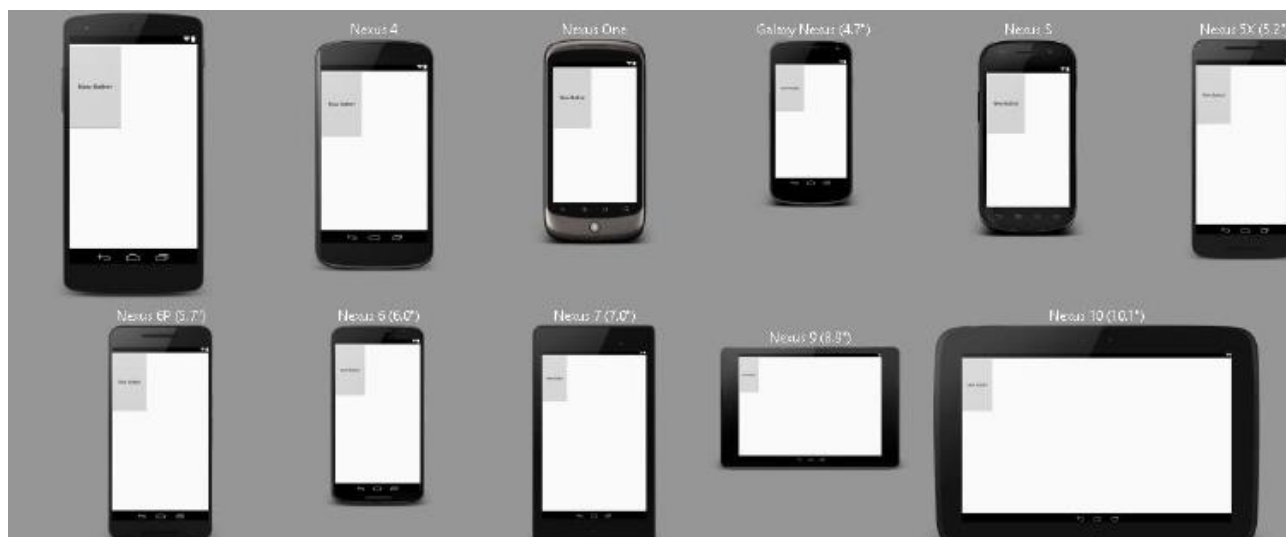


Рис. 4

Але при завданні розмірів кнопки 150x200 px маємо:

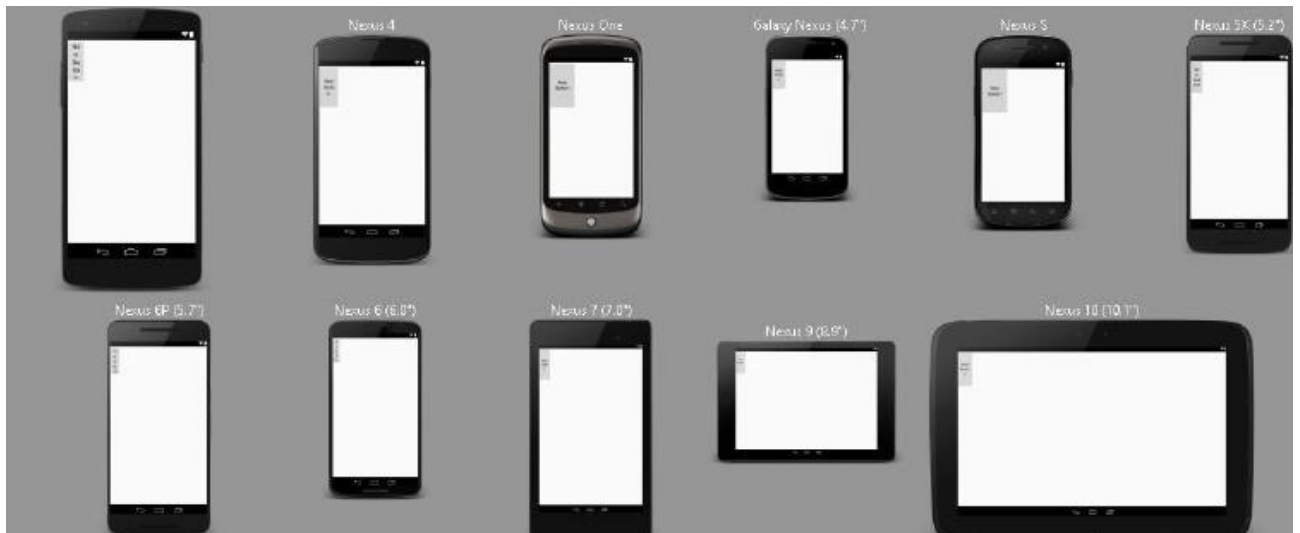


Рис. 5

Властивість **weight** позначає долю екрана, яку буде займати віджет. Наприклад, дві кнопки з такими параметрами:

```

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Button"
    android:id="@+id/button"
    android:layout_weight="0" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Button"
    android:id="@+id/button2"
    android:layout_weight="2" />

```

Рис. 6

На екрані розташуються наступним чином:

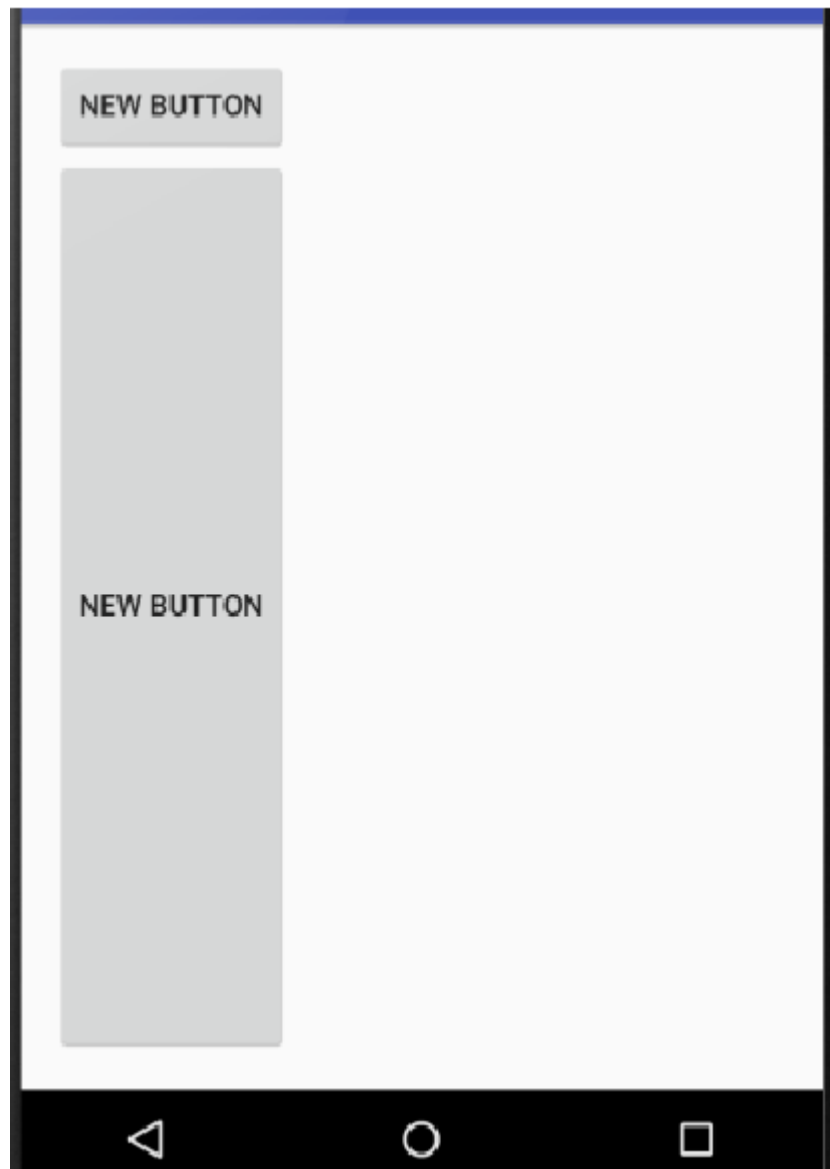


Рис. 7

Ідентифікатор віджета генерується Android Studio автоматично, та має спеціальний формат, наприклад: "`@+id/button2`".

Властивість **text** містить текст, що відображається на віджеті, наприклад, назва кнопки. При присвоєнні властивості **text** строкового значення:

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="New Button1"
```

Рис. 8

Android Studio видає попередження: **Hardcoded string "New Button1", should use @string resource**. Тобто, при завданні строкових значень, слід визначати строки в відповідному файлі ресурсу:

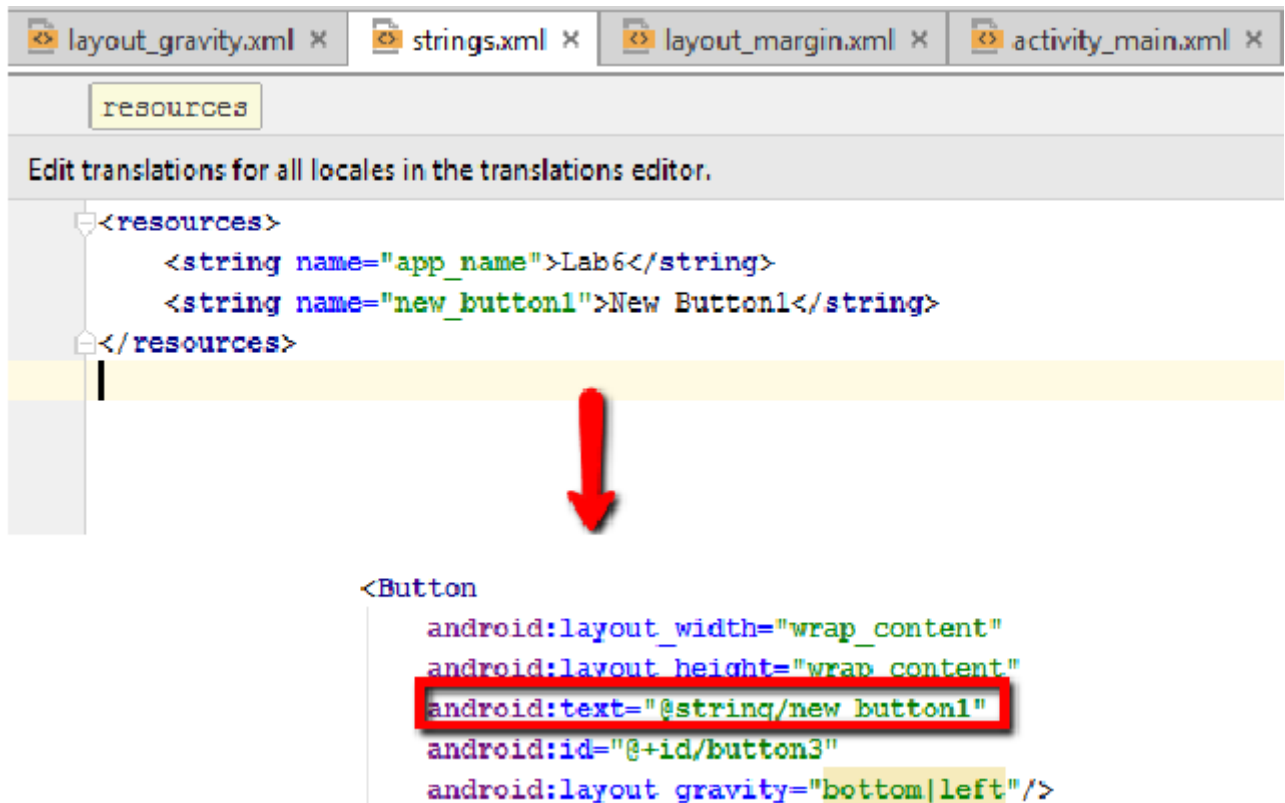


Рис. 9

Властивості **padding** та **margin** визначають відстані. Властивість **padding** визначає відстань від границі віджета до початку вмісту. Властивість **margin** – це відстань зовні від границі віджета до інших віджетів (рис. 10).

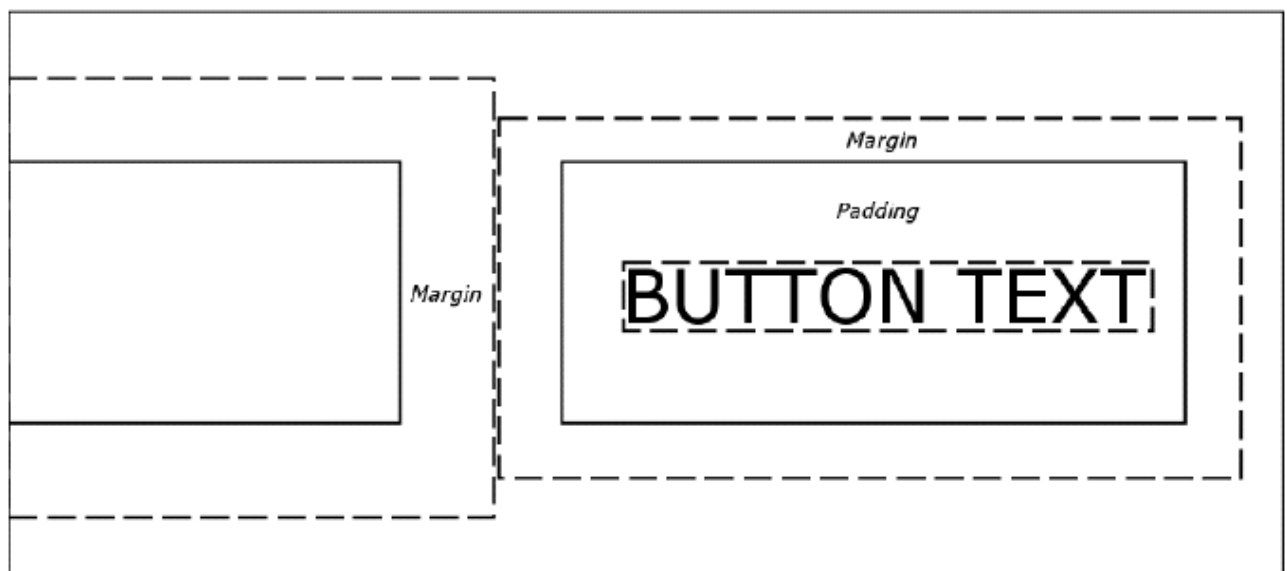


Рис. 10

Властивість **gravity** визначає розташування віджета на activity. Наприклад, визначення властивості gravity п'яти кнопок у XML файлі та відповідне розташування на activity додатку:

```

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/new_button1"
    android:id="@+id/button3"
    android:layout_gravity="bottom|left"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Button4"
    android:id="@+id/button4"
    android:layout_gravity="bottom|right"/

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Button3"
    android:id="@+id/button5"
    android:layout_gravity="center"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Button"
    android:id="@+id/button15"
    android:layout_gravity="top|left"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Button"
    android:id="@+id/button16"
    android:layout_gravity="top|right"/>

```

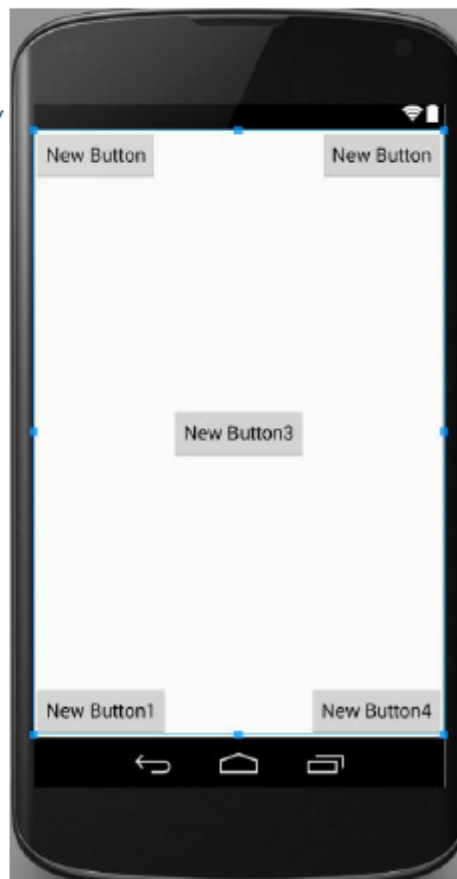


Рис. 11

Обробка натиснень

При обробці натиснення кнопки чи іншого об'єкта View можна застосувати декілька підходів.

Визначення атрибута **android:onClick**. Для цього в xml відповідного layout в секції <Button> потрібно визначити атрибут android:onClick. Значення цього атрибута повинно бути ім'я метода, який буде оброблювати натиснення (рис. 12).


```

<?xml version="1.0" encoding="utf-8"?>
<Button xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/button_send"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_send"
    android:onClick="sendMessage" />

```

Рис. 12

Відповідно java-клас макет повинен містити реалізацію метода обробки натиснення (рис. 13).

```

/** Called when the user touches the button */
public void sendMessage(View view) {
    // Do something in response to button click
}

```

Рис. 13

Використання **OnClickListener**. Для використання обробника події (event handler) створюється об'єкт **View.OnClickListener**, який асоціюється з віджетом за допомогою метода **setOnClickListener(View.OnClickListener)** (рис. 14).

```

Button button = (Button) findViewById(R.id.button_send);
button.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        // Do something in response to button click
    }
});

```

Рис. 14

Наприклад, обробка натиснення трьох кнопок та вивід у TextView імені натиснутої кнопки може бути виконана наступним чином

```

View.OnClickListener onClickListener = new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        switch (v.getId()){
            case R.id.button: myTextView.setText(R.string.str_btn1);
break;
            case R.id.button2: myTextView.setText(R.string.str_btn2);
break;
            case R.id.button3: myTextView.setText(R.string.str_btn3);
break;
            case R.id.textView: myTextView.setText(R.string.str_txtvwr);
break;
        }
    }
};

button1.setOnClickListener(onClickListener);
button2.setOnClickListener(onClickListener);
button3.setOnClickListener(onClickListener);

```

Рис. 15

Аналогічним чином оброблюється натиснення інших об'єктів View, для яких ввімкнена властивість **android:clickable**.

Підчас роботи додатку у java-класах можливо змінювати атрибути елементів UI та викликати їх специфічні методи.

Наприклад, доступ до деяких властивостей віджетів TextView, Button, CheckBox продемонстровано нижче

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    TextView myTextVeiw= (TextView) findViewById(R.id.myText);
    myTextVeiw.setText("Lab 7.Test");
    Button myButton=(Button) findViewById(R.id.myButton);
    myButton.setText("My new button");
    CheckBox mycheckBox = (CheckBox) findViewById(R.id.mycheckBox);
    mycheckBox.setChecked(true);
}

```

Рис. 16

Більш детально роботу з елементами Button викладено

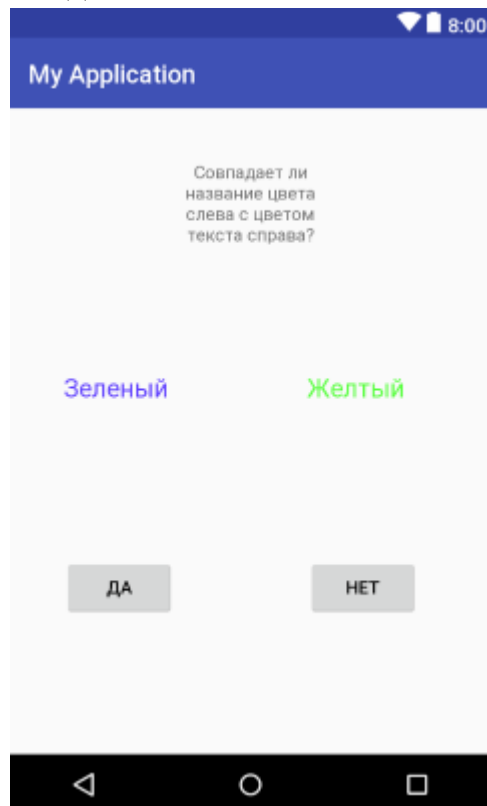
<https://developer.android.com/reference/android/widget/Button.html>

Більш детально роботу з елементами TextView викладено

<https://developer.android.com/reference/android/widget/TextView.html>

Завдання до лабораторної роботи

1. Створити додаток у Android Studio з одним activity на якому реалізувати варіант логічної гри на визначення співпадіння кольору тексту та назви кольору. Можливий варіант розташування елементів TextView та Button наведено на макеті нижче



2. Реалізувати генерацію випадковим чином кольорів та підрахунок правильних відповідей впродовж 1 хвилини та вивід результатів роботи програми.

Контрольні запитання

1. Які основні типи віджетів вам відомі?
2. Які основні способи обробки натиснення кнопки?
3. Які специфічні атрибути та методи визначені для елемента TextView?