

Лабораторна робота №4. Робота з компонентами UI

Мета: вивчити порядок роботи з найбільш поширеними віджетами.

Теоретичні відомості

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

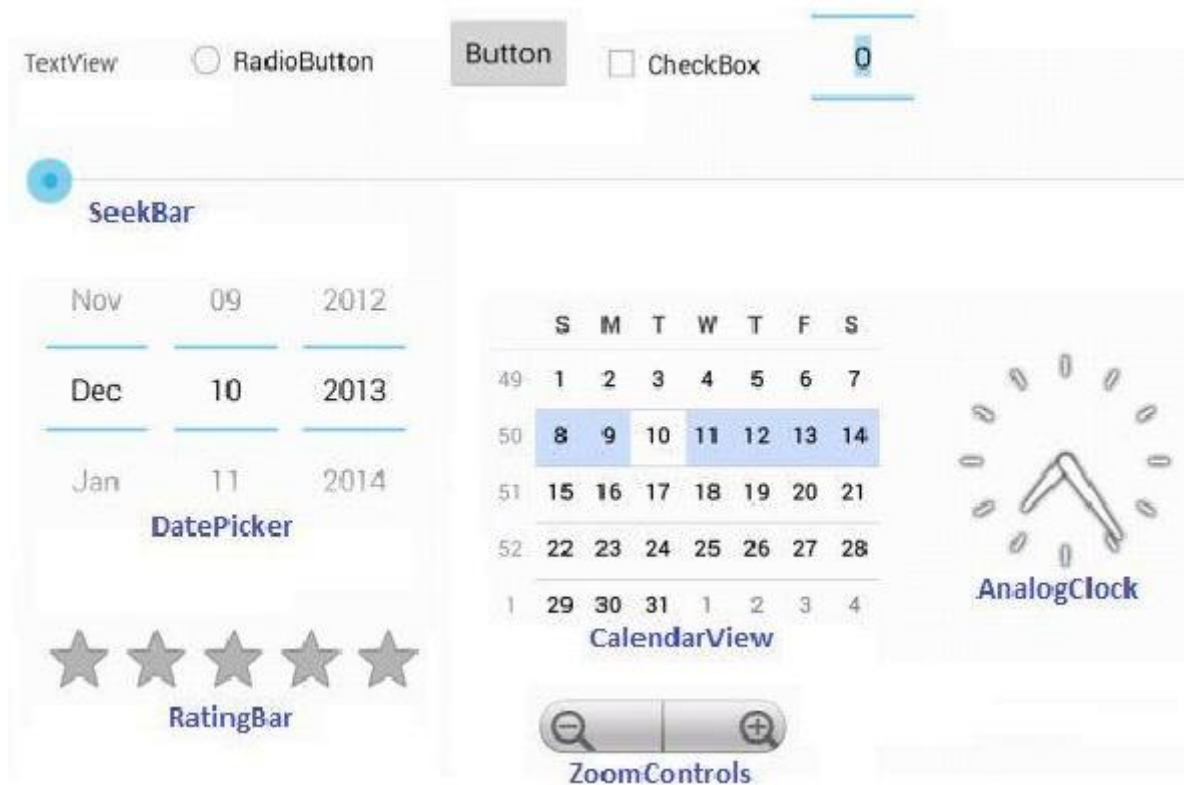


Рис. 1

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

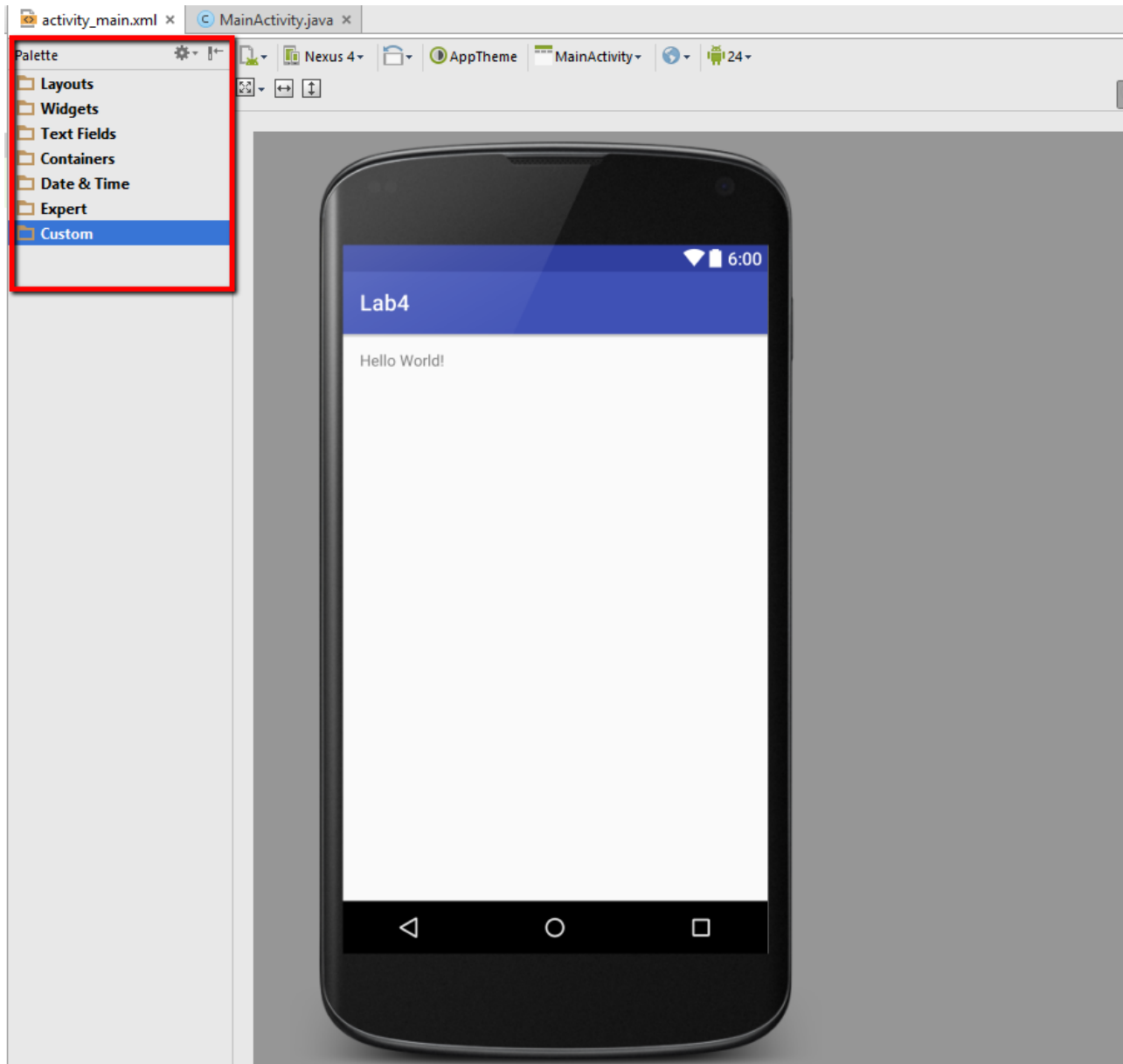


Рис. 2

Всі елементи графічного інтерфейса в додатку Android створюються за допомогою об'єктів **View** (віджет) і **ViewGroup**. Об'єкти **View** формують на екрані елементи з якими буде взаємодіяти користувач. Об'єкти **ViewGroup** містять інші об'єкти **View** та **ViewGroup** і визначають інтерфейс користувача. Найбільш поширений тип **ViewGroup** – це **layout**. В XML файлі кожного **layout** визначаються всі елементи **View**, їх розташування та властивості.

Зовнішній вигляд всіх віджетів можна визначати двома способами:

- У вікні Properties при виділеному віджеті:

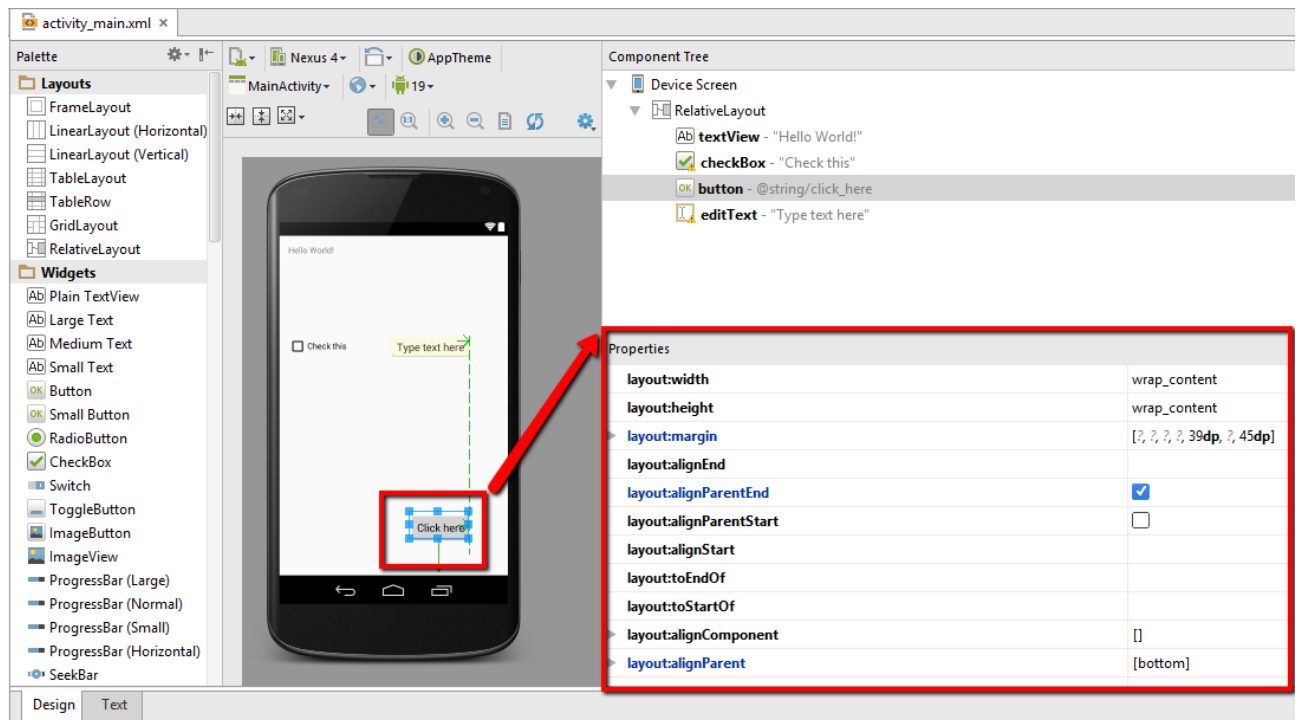


Рис. 3

- У XML файлі відповідного layout.

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 визначенні 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:

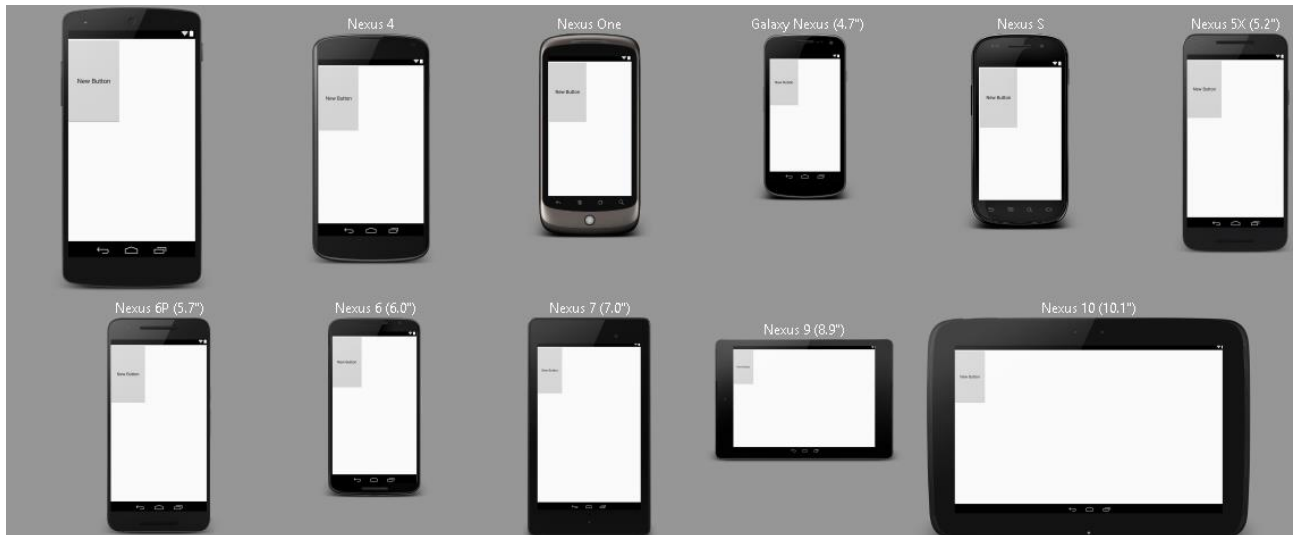


Рис. 4

Але при завданні розмірів кнопки 150x200 px маємо:

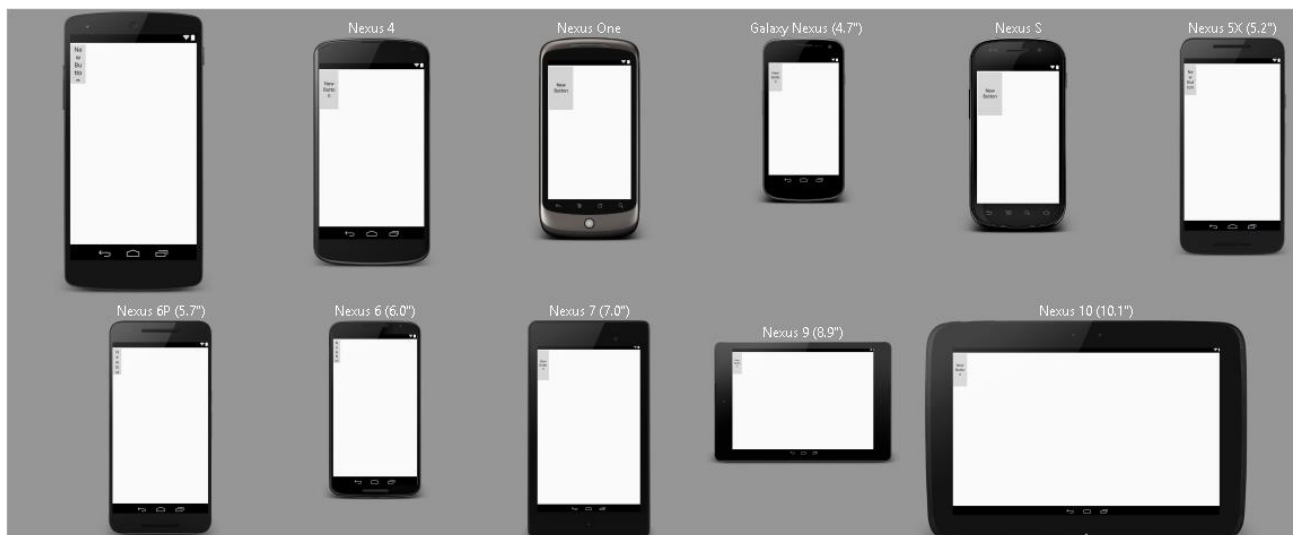


Рис. 5

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:

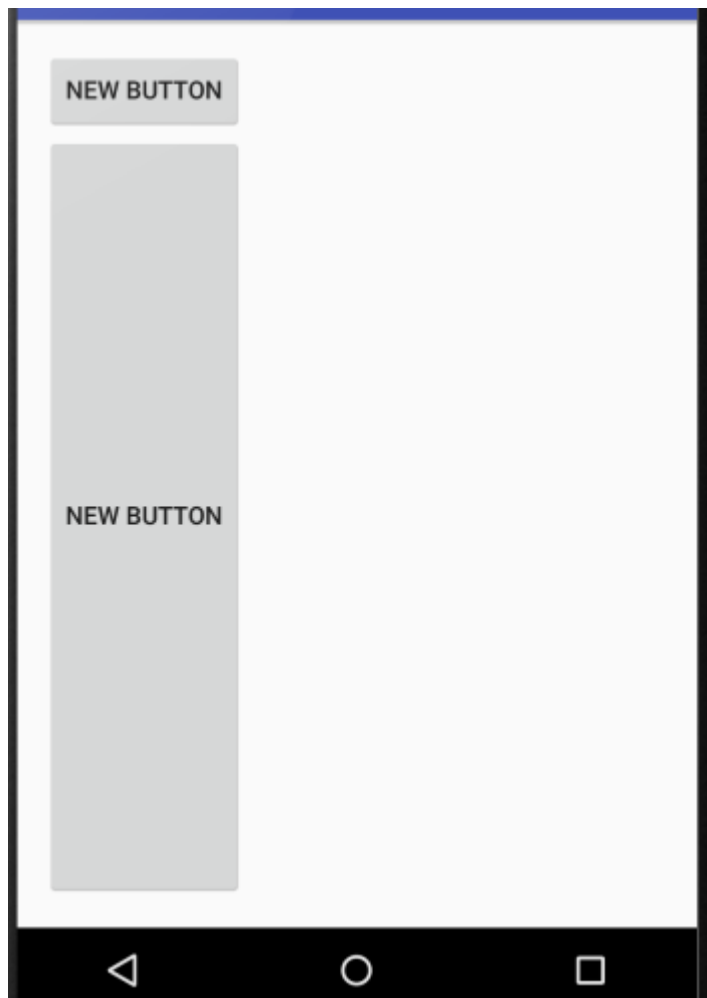


Рис. 7

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:

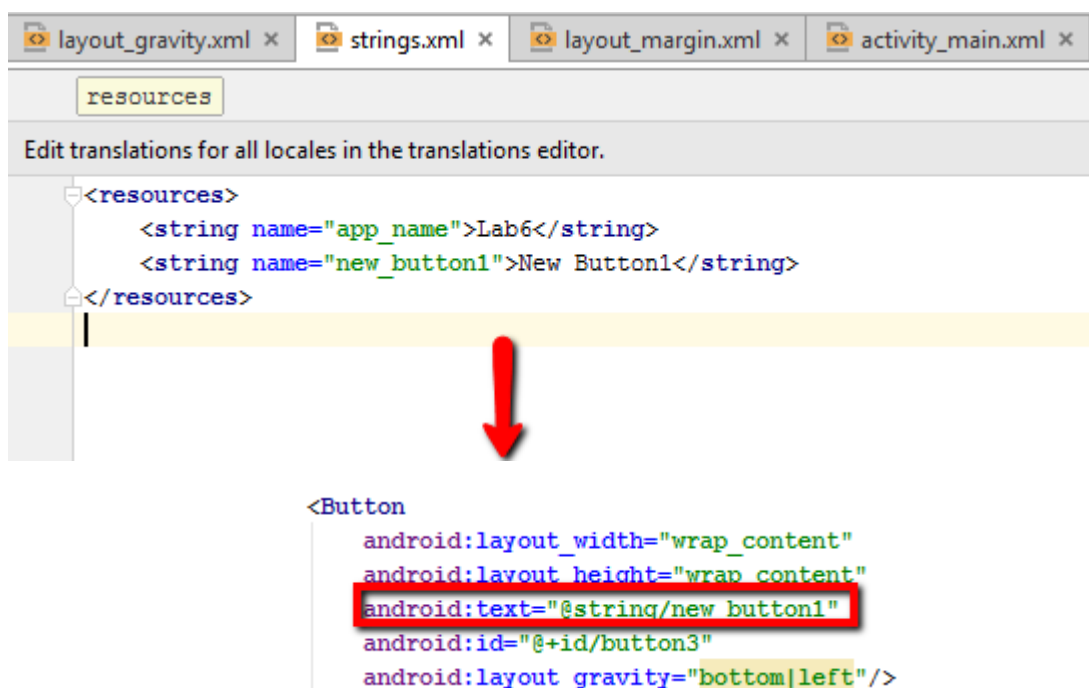


Рис. 9

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

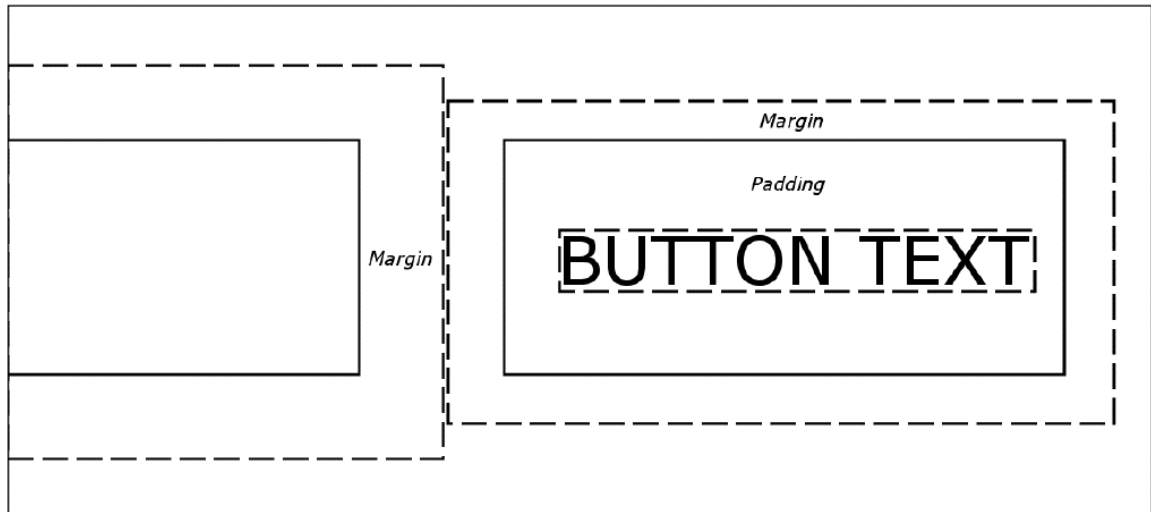


Рис. 10

Можна задавати відстані у всіх можливих напрямках рівними:

```
...
android:layout_margin="43dp"
android:padding="10dp"
...
```

Рис. 11

Або вказувати конкретний напрям:

```
android:layout_marginTop="43dp"
android:layout_marginBottom="43dp"
android:paddingLeft="5dp"
android:paddingRight="5dp"
```

Рис. 12

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"/>

```

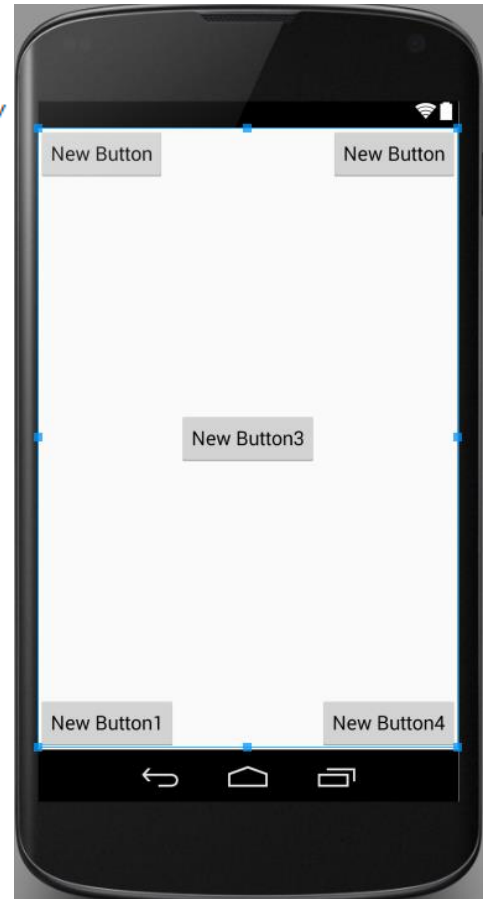


Рис. 13

Завдання до лабораторної роботи

1. Створіть новий проект з **Blank Activity**.
2. Створіть кнопку в **MainActivity**.
3. В меню **Properties** створеної кнопки визначте властивість **text** як **Left Button**.
4. Властивість **width=150dp**.
5. Властивість **layout:margin** визначте згідно рис. 14.

| | |
|------------------------|------------------------------------|
| ▼ layout:margin | [?, 10dp, 100dp, 50dp, 50dp, ?, ?] |
| all | |
| left | 10dp |
| top | 100dp |
| right | 50dp |
| bottom | 50dp |

Рис. 14

6. Створіть другу кнопку та розташуйте справа від першої.
7. Властивість **text= Right**, **width= 80dp**.
8. Перетягніть на макет віджет **Plain TextView** та розташуйте його знизу кнопок та по центру.
9. Визначте властивість **textSize= 65sp**, **layoutWidth=match_parent**, **gravity=center**, **alpha=0.5**.
10. Виберіть колір у властивості **background**.
11. Визначте колір тексту у властивості **textColor**.
12. Властивість **typeface** змініть на **serif**.
13. Перетягніть віджет **ImageView** та розташуйте його нижче **TextView**.
14. У вікні **Properties** визначте властивість **src** для відображення картинки, наприклад, **ic_launcher**.
15. Перегляньте файл **MainActivity.xml** та зазначте, яким чином вказані у вікні **Properties** властивості записуються у форматі XML.

Контрольні запитання

1. Які основні типи віджетів вам відомі?
2. Які основні властивості віджетів ви знаєте?