

## Лабораторна робота №5. Робота з різними типами Layout

**Мета:** вивчити порядок роботи з різними типами макетів.

### Теоретичні відомості

**Layout** представляє собою структуру, яка групує об'єкти графічного інтерфейсу. Android пропонує декілька типів Layout, які відрізняються розташуванням компонентів (рис. 1).

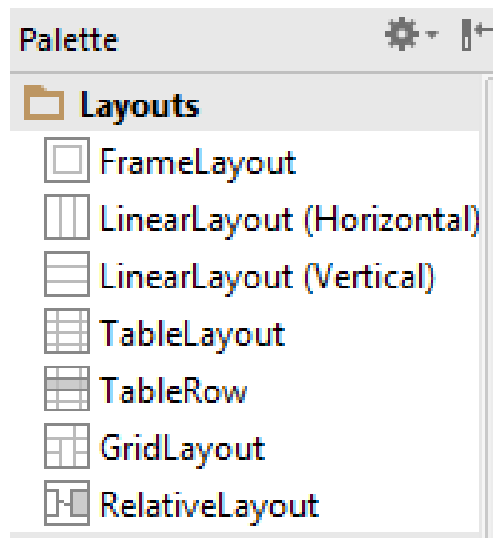


Рис. 1

За замовченням, при створенні нового проекту визначається тип **RelativeLayout**.

**RelativeLayout** – це такий тип макету, в якому всі елементи розташовуються на відносних один до одного позиціях. Наприклад, один віджет може розташовуватися зліва чи знизу порівняно з іншим віджетом.

Деякі властивості позицій, які можуть визначатися при створенні RelativeLayout наведені нижче.

Розташування відносно батьківського елемента (значенням атрибуту є true/false):

- **android:layout\_alignParentTop** – верхня границя елемента розташовується на верхній границі RelativeLayout;
- **android:layout\_alignParentLeft** – ліва границя елемента розташовується на лівій границі RelativeLayout;
- **android:layout\_alignParentRight** – права границя елемента розташовується на правій границі RelativeLayout;

- **android:layout\_alignParentBottom** – нижня границя елемента розташовується на нижній границі RelativeLayout;
- **android:layout\_centerInParent** – елемент розташовується в центрі батьківського по горизонталі и вертикалі;
- **android:layout\_centerHorizontal** – елемент розташовується в центрі батьківського по горизонталі;
- **android:layout\_centerVertical** – елемент розташовується в центрі батьківського по вертикалі.

Розташування відносно інших елементів (значенням атрибуту є id іншого елемента):

- **android:layout\_above** – елемент розташовується зверху від іншого елемента;
- **android:layout\_toLeftOf** – елемент розташовується ліворуч від іншого елемента;
- **android:layout\_toRightOf** – елемент розташовується праворуч від іншого елемента;
- **android:layout\_below** – елемент розташовується знизу від іншого елемента;
- **android:layout\_alignBaseline** – базова лінія елемента вирівнюється по базовій лінії іншого елемента;
- **android:layout\_alignTop** – верхня границя елемента вирівнюється по верхній границі іншого елемента;
- **android:layout\_alignLeft** – ліва границя елемента вирівнюється по лівій границі іншого елемента;
- **android:layout\_alignRight** – права границя елемента вирівнюється по правій границі іншого елемента;
- **android:layout\_alignBottom** – нижня границя елемента вирівнюється по нижній границі іншого елемента.

Приклад xml файла RelativeLayout:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp" >
    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/reminder" />
    <Spinner
        android:id="@+id/dates"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:layout_alignParentLeft="true"
        android:layout_toLeftOf="@+id/times" />
    <Spinner
        android:id="@id/times"
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:layout_alignParentRight="true" />
    <Button
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/times"
        android:layout_alignParentRight="true"
        android:text="@string/done" />
</RelativeLayout>

```

Рис. 1

**LinearLayout.** Макет для одного чи декількох елементів в лінію, горизонтально або вертикально. Для вибору орієнтації використовується атрибут **android:orientation** з можливими значеннями «horizontal»\«vertical» (рис. 2).

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="com.example.alex.lab6.MainActivity"
    android:orientation="vertical">

    <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" />

</LinearLayout>
```

Рис. 2

**TableLayout.** Макет для розташування елементів у вигляді таблиці. Ряди таблиці задаються за допомогою елемента TableRow (тег TableRow в xml). Кількість стовпців в таблиці дорівнює максимальній кількості стовпців у рядках. Комірки таблиці можна залишати пустими.

Приклад. xml файл, в якому визначається таблиця 3x3 по 3 кнопки в ряд (рис. 3).

```
<?xml version="1.0" encoding="utf-8" ?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <Button
            android:text="New Button"
            android:id="@+id/button6"
            android:layout_column="0" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="New Button"
            android:id="@+id/button7"
```

```

        android:layout_column="1" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Button"
        android:id="@+id/button8"
        android:layout_column="2" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:text="New Button"
        android:id="@+id/button9"
        android:layout_column="0" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Button"
        android:id="@+id/button10"
        android:layout_column="1"
        android:layout_marginLeft="10dp"
        android:paddingTop="20dp"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Button"
        android:id="@+id/button11"
        android:layout_column="2" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:text="New Button"
        android:id="@+id/button12"
        android:layout_column="0" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Button"
        android:id="@+id/button13"
        android:layout_column="1" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Button"
        android:id="@+id/button14"
        android:layout_column="2" />
</TableRow>
</TableLayout>

```



Рис. 3

**FrameLayout.** Макет для відображення одного елемента. Всі елементи, які додаються на макет розташовуються у верхньому лівому куті екрана .

Приклад.

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_gravity="center_horizontal">

        <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"/>
</FrameLayout>
</LinearLayout>

```

### Створення layout для landscape\portrait режимів.

Для створення представлення для **landscape** орієнтації потрібно створити у папці **res** нову папку з назвою **layout-land** (рис. 4), в яку покласти xml файл з тією ж назвою, що и для основного portrait режиму.

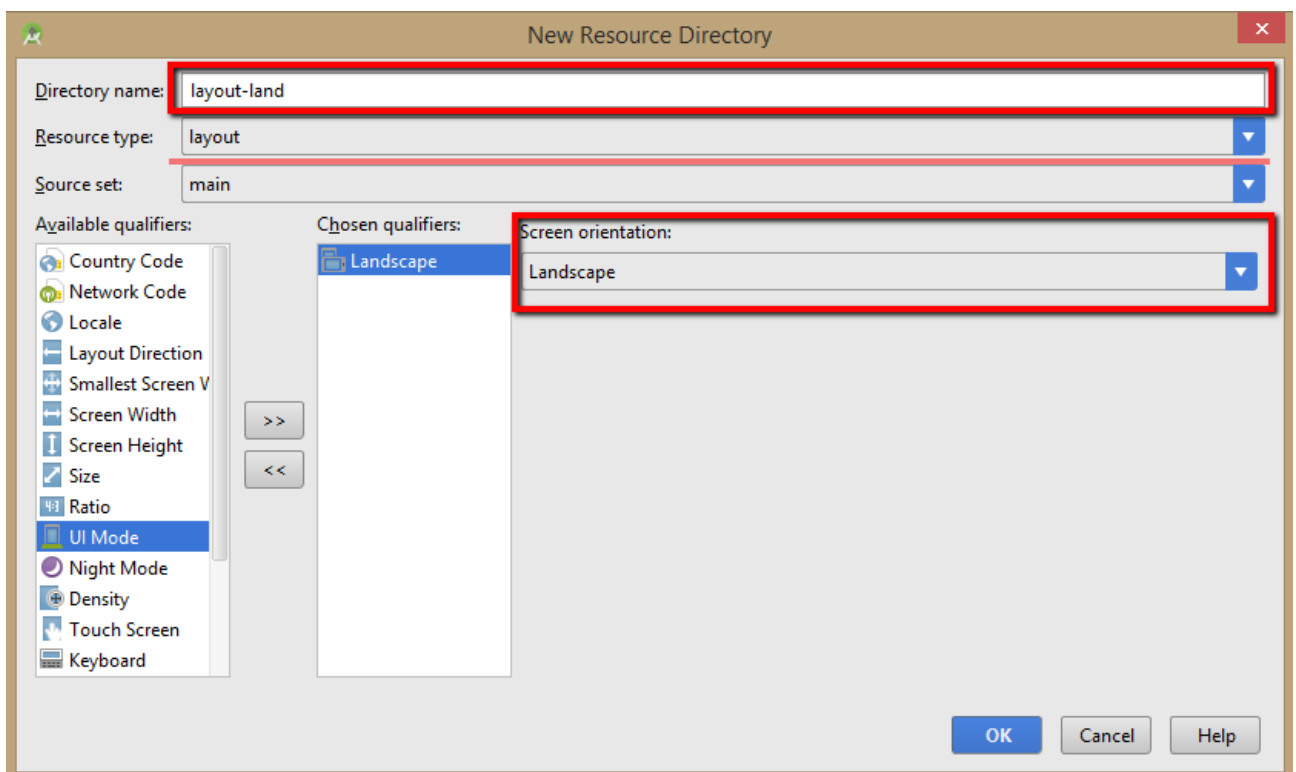


Рис. 4

Для того, щоб побачити створену папку, потрібно переключитись в режим відображення структури Project (рис. 5).

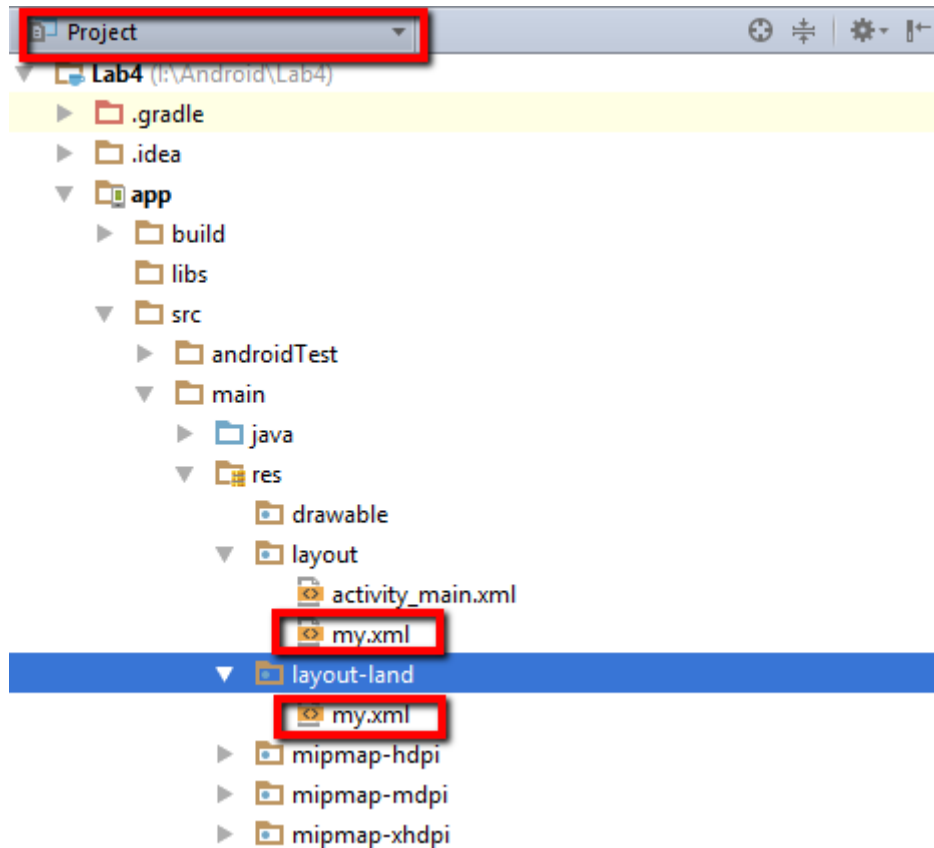


Рис. 5

### Завдання до лабораторної роботи

1. Створити activity для кожного з розглянутих layout. Розташувати на них віджети та використати для вирівнювання специфічні для кожного макета атрибути.
2. Створіть нову activity з LinearLayout (горизонтальним) макетом. У ньому створіть два вкладених макета: LinearLayout (вертикальний), LinearLayout (горизонтальний) (рис. 6). При цьому, макети далі описуються для landscape орієнтації.

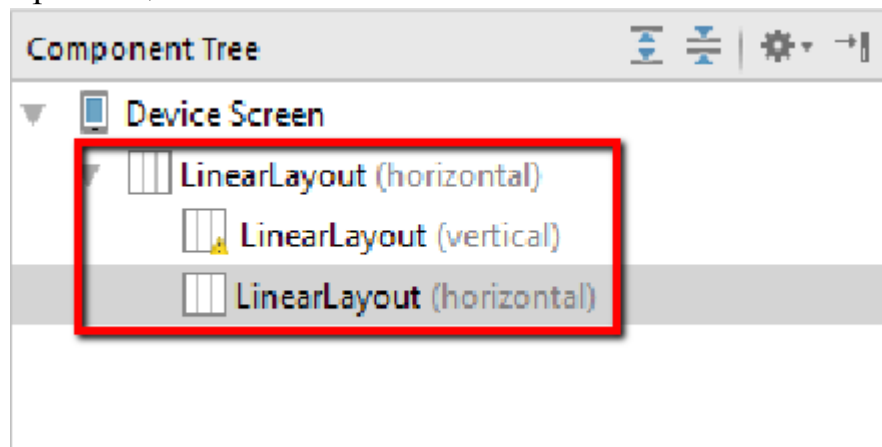
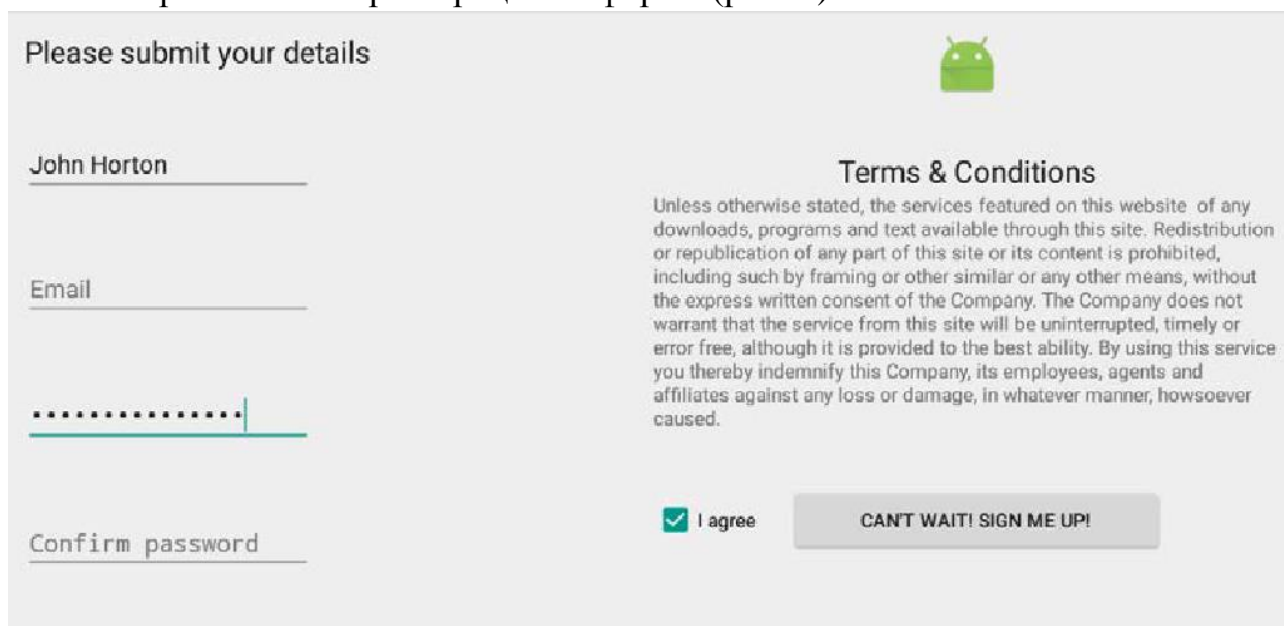


Рис. 6

3. Оберіть перший (вертикальний) LinearLayout та визначте його атрибут **layout:weight=40**. Встановіть колір фону у атрибуті **background**.



4. Для другого (горизонтального) `LinearLayout` визначте атрибут `layout:weight=60`.
5. У менший `LinearLayout` (40%) додайте віджет `ScrollView`.
6. Додайте вертикальний `LinearLayout` на створений `ScrollView`.
7. На створений на попередньому етапі `LinearLayout` додайте близько 10 віджетів `Large Text`.
8. На більший `LinearLayout` (60%) додайте елемент `ImageView`.
9. У `ImageView` відобразіть яку-небудь картинку, наприклад, `ic_launcher`, яка вже знаходиться у папці `res`.
10. Встановіть властивість `layout:weight=1` елемента `ImageView`.
11. Розробіть макет реєстраційної форми (рис. 7).



Please submit your details

John Horton

Email

.....|

Confirm password

Terms & Conditions

Unless otherwise stated, the services featured on this website of any downloads, programs and text available through this site. Redistribution or republication of any part of this site or its content is prohibited, including such by framing or other similar or any other means, without the express written consent of the Company. The Company does not warrant that the service from this site will be uninterrupted, timely or error free, although it is provided to the best ability. By using this service you thereby indemnify this Company, its employees, agents and affiliates against any loss or damage, in whatever manner, howsoever caused.

I agree

CANT WAIT! SIGN ME UP!

Рис. 7

### Контрольні запитання

1. Які типи макетів ви знаєте? У чому різниця між ними?
2. Назвіть специфічні атрибути `RelativeLayout`.
3. Яким чином визначити макет для різних орієнтацій екрана?