

L4. Activity. Створюємо інтерактивний додаток

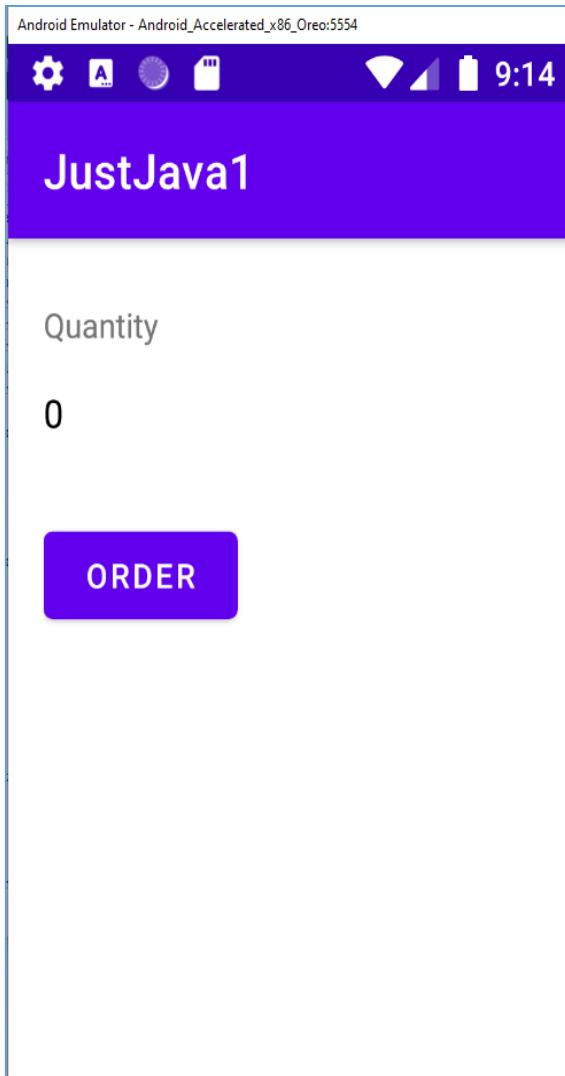
На прикладі програми *Just Java* вивчимо основи програмування і мови *Java*, а також розглянемо підходи до налагодження і виправлення помилок в кодї

Application Name
Package Name
Use template
Minimum SDK Level

Just Java
android.example.justjava
Empty Activity
API16



Л4. Activity. Створюємо інтерактивний додаток. Етап 1



Крок 1. Вибираємо Views (*TextView, TextView, Button*)

Крок 2. Позиція Views (*LinearLayout*)

Крок 3. Стель Views (колір та розмір шрифту)

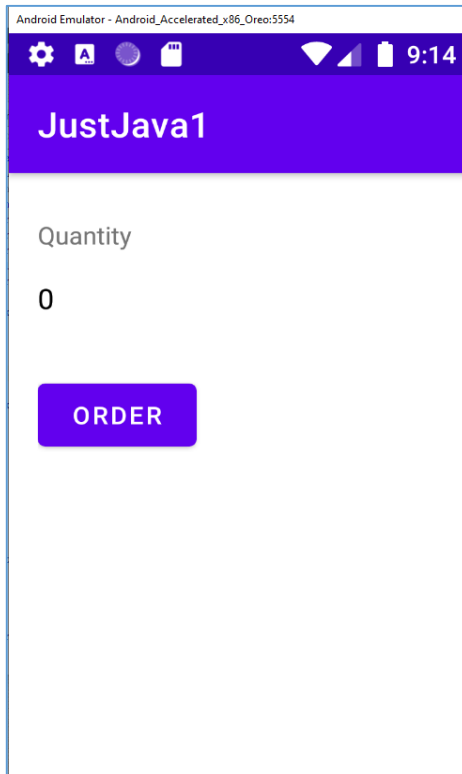
Переформатувати та впорядкувати текст коду виконується з меню **Code** → **Reformat Code** або натиснення **Ctrl+Alt+L**.

Крок 4. Переходимо до файлу **activity_main.xml** та створюємо макет

Крок 5. *TextView*, що відображає 0 додаємо ID name – **@+id/quantity_text_view**

Крок 6. Відступи між елементами **16dp**. Шрифт (0) має бути – black **#000000** та розмір – **16sp**

Л4. Activity. Створюємо інтерактивний додаток. Етап 1



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
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"
android:orientation="vertical"
tools:context=".MainActivity"
android:padding="16dp">

<TextView
    android:id="@+id/Quan"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="10sp"
    android:text="Quantity" />

<TextView
    android:id="@+id/quantity_text_view"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:layout_marginTop="16sp"
    android:layout_marginBottom="16sp"
    android:textColor="#000000"
    android:text="0"/>

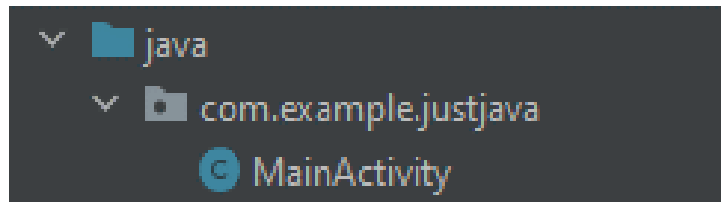
<Button
    android:id="@+id/order_botton_view"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16sp"
    android:text="ORDER"
    android:onClick="submitOrder"/>
</LinearLayout>
```

Л4. Activity. Створюємо інтерактивний додаток. Етап 1

Крок 7. Кожна активність спрямована на певну дію користувача. Наприклад, розглянемо додаток Sunshine.



Основна активність визначена у файлі `MainActivity.java`.



Л4. Activity. Створюємо інтерактивний додаток. Етап 1

Переходимо до файлу **MainActivity.java** та замінюємо код, що студія згенерувала на код

```
package com.example.android.justjava;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import android.view.View;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    /**
     * This method is called when the order button is clicked.
     */
    public void submitOrder(View view) {
        display(1);
    }

    /**
     * This method displays the given quantity value on the screen.
     */
    private void display(int number) {
        TextView quantityTextView = (TextView)
            findViewById(R.id.quantity_text_view);
        quantityTextView.setText("" + number);
    }
}
```

Л4. Activity. Створюємо інтерактивний додаток. Етап 1

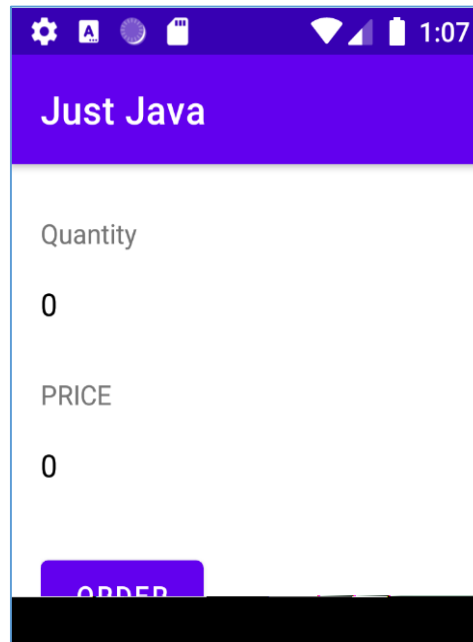
Крок 8. Додаємо атрибут кнопці. Для того щоб кнопка щось робила в XML коді потрібно додати атрибут `android:onClick="submitOrder"`

Коли нажимаємо кнопку система намагається виконати метод `submitOrder` з файлу `MainActivity.java`. Кожний метод має фігурні дужки. Також можемо побачити ще метод `onCreate, display`. Так у коді метод `display(1)` має відобразити 1 на екрані.

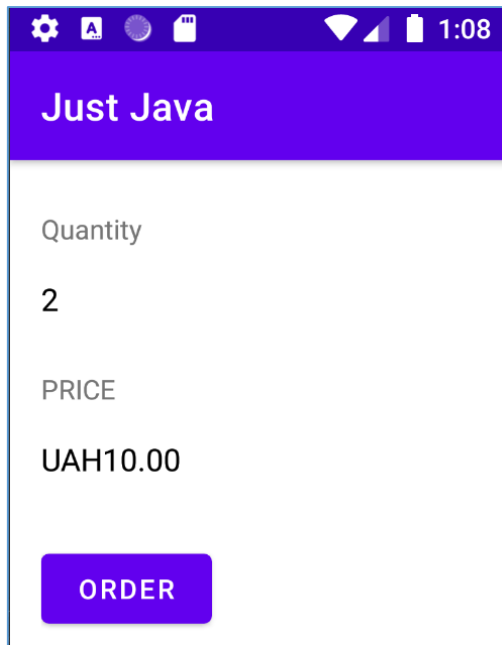
Крок 9. Запускаємо додаток і дивимось, що відбувається (має появиться 1)

Крок 10. Вносимо зміни у XML. Оновлюємо макет.

Елементу `TextView`, що відображає ціну задаємо ідентифікатор `@+id/price_text_view`. Інтервали між елементами має бути 16 dp.



Л4. Activity. Створюємо інтерактивний додаток. Етап 1



Крок 11. Додаємо в файл `MainActivity.java` новий метод **`displayPrice()`**

```
private void displayPrice(int number) {  
    TextView priceTextView = (TextView) findViewById(R.id.price_text_view);  
    priceTextView.setText(NumberFormat.getCurrencyInstance().format(number));  
}
```

Корегуємо код, додавши український формат UAN

```
private void displayPrice(int number) {  
    TextView priceTextView = (TextView) findViewById(R.id.price_text_view);  
    priceTextView.setText(java.text.NumberFormat.getCurrencyInstance(new  
        Locale("ua", "UA")).format(number));  
}
```

Переконайтеся, що автоімпорт включений. Для Windows зайдіть в **File>Settings>Editor>General>Auto Import** та додайте всі наступні опції:

- *Show import popup*
- *Optimize imports on the fly*
- *Add unambiguous imports on the fly*
- *Insert imports on paste → All*

Крок 12. Корегуємо метод **`submitOrder`** та додаємо в ньому рядок **`displayPrice(2*5)`**; (5 це ціна 1 кави)

```
public void submitOrder(View view) {  
    display(2);  
    displayPrice(2*5);  
}
```

Л4. Activity. Створюємо інтерактивний додаток. Етап 1

Крок 13. Тестуємо додаток

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".MainActivity"
    android:padding="16dp">

    <TextView
        android:id="@+id/Quan"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10sp"
        android:text="Quantity" />

    <TextView
        android:id="@+id/quantity_text_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:layout_marginTop="16sp"
        android:layout_marginBottom="16sp"
        android:textColor="#000000"
        android:text="0"/>

    <TextView
        android:id="@+id/Price"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10sp"
        android:layout_marginBottom="16sp"
        android:text="PRICE" />

    <TextView
        android:id="@+id/price_text_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:layout_marginBottom="16sp"
        android:textColor="#000000"
        android:text="0"/>

    <Button
        android:id="@+id/order_botton_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16sp"
        android:text="ORDER"
        android:onClick="submitOrder"/>

</LinearLayout>
```


Л4. Activity. Створюємо інтерактивний додаток. Етап 1

MainActivity.java

```
package com.example.justjava;

import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
import java.util.Locale;

//This app displays an order form to order coffee.
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    // This method is called when the order button is clicked.
    public void submitOrder(View view) {
        display(2);
        displayPrice(2*5);
    }
    // This method displays the given quantity value on the screen.
    private void display(int number) {
        TextView quantityTextView = (TextView) findViewById(R.id.quantity_text_view);
        quantityTextView.setText("" + number);
    }
    // This method displays the given price on the screen.
    private void displayPrice(int number) {
        TextView priceTextView = (TextView) findViewById(R.id.price_text_view);
        priceTextView.setText(java.text.NumberFormat.getCurrencyInstance(new
        Locale("ua", "UA")).format(number));
    }
}
```

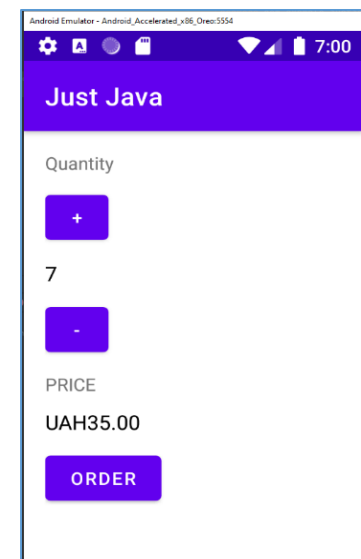
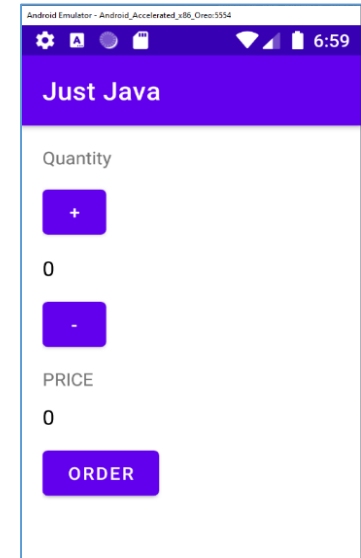
Л4. Activity. Створюємо інтерактивний додаток. Етап 1

Крок 14. Додаємо змінну, яка буде зберігати кількість чашок кофе у метод **submitOrder**

```
public void submitOrder(View view) {  
    int numberOfCofees = 2;  
    display(numberOfCofees);  
    displayPrice (numberOfCofees *5);  
}
```

При запуску *Android Studio* в нижній частині можемо побачити вікно **Android DDMS**. Це інструмент для налагодження, він називається **DDMS** (Davlik Debug Monitor Server). Він веде системний лог (logcat). Коли в телефоні відбувається збій в логах зберігається опис того, що телефон збирається зробити.

Крок 15. Додаємо кнопки + та - , щоб збільшувати та зменшувати заказ



Л4. Activity. Створюємо інтерактивний додаток. Етап 1

1) Для цього додаємо в розмітку кнопки + та мінус. Висота та ширина кнопок 48dp. При натисненні кнопки + в **mainActivity** має викликатись метод **increment** (при натисненні - **decrement**).

<Button

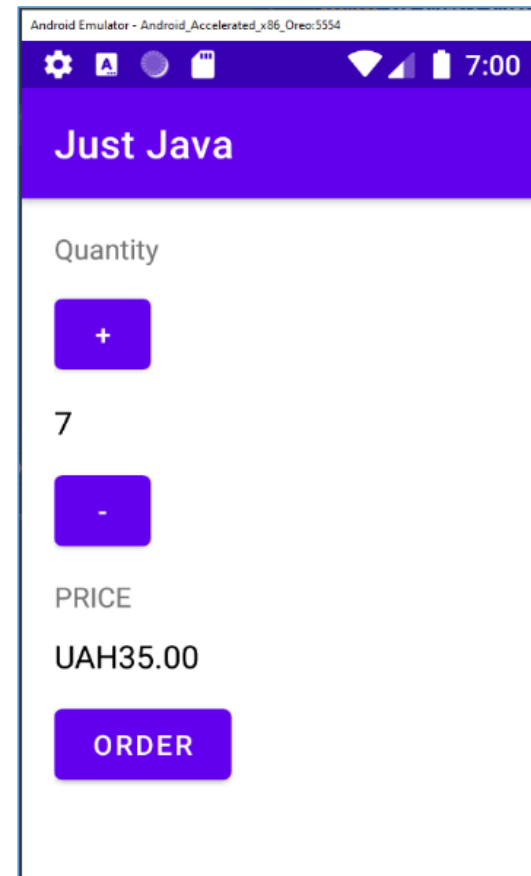
```
android:id="@+id/plus_botton_view"  
android:layout_width="48dp"  
android:layout_height="48dp"  
android:layout_marginBottom="10dp"  
android:onClick="countPlus"  
android:text="+" />
```

<TextView

```
android:id="@+id/quantity_text_view"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginBottom="10dp"  
android:text="0"  
android:textColor="#000000"  
android:textSize="16sp" />
```

<Button

```
android:id="@+id/minus_botton_view"  
android:layout_width="48dp"  
android:layout_height="48dp"  
android:layout_marginBottom="10dp"  
android:onClick="countMinus"  
android:text="-" />
```



Л4. Activity. Створюємо інтерактивний додаток. Етап 1

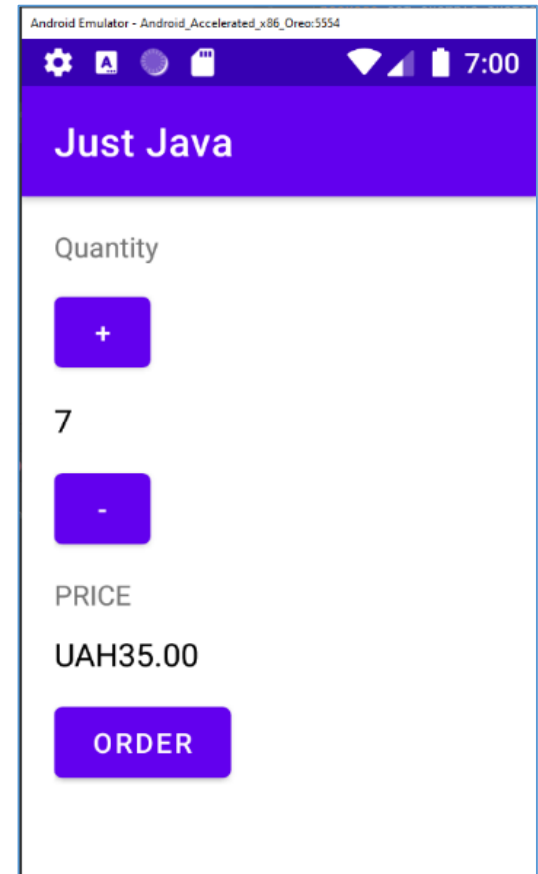
2) Створюємо глобальну змінну **int numberOfCofees = 0**. У методі **increment / decrement** збільшуємо /зменшуємо змінну та відображаємо кількість на екрані.

```
int numberOfCofees = 0;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}

public void submitOrder(View view) {
    displayPrice(numberOfCofees * 5);
}

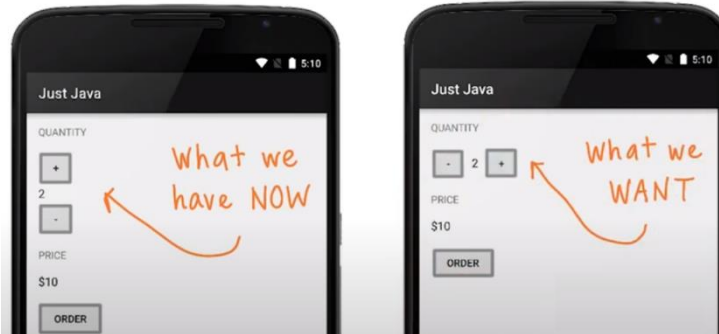
public void countPlus(View view) {
    numberOfCofees++;
    display(numberOfCofees);
}

public void countMinus(View view) {
    if(numberOfCofees>1) numberOfCofees--;
    display(numberOfCofees);
}
```



Л4. Activity. Створюємо інтерактивний додаток. Етап 1

Крок 16. Кнопки працюють але макет потребує вдосконалення. Потрібно краще використовувати горизонтальну площу екрану.



Для побудови макета необхідно 3 кроки.

- 1) Вибрати Views. Які потрібні?
- 2) Position Views. Які ViewGroups розташовані на екрані?
- 3) Style Views. Чи потрібно щось зробити, щоб візуально змінити ці елементи

Необхідно 2 кнопки та TextView з кількістю кави помістити в LinearLayout з горизонтальною орієнтацією.

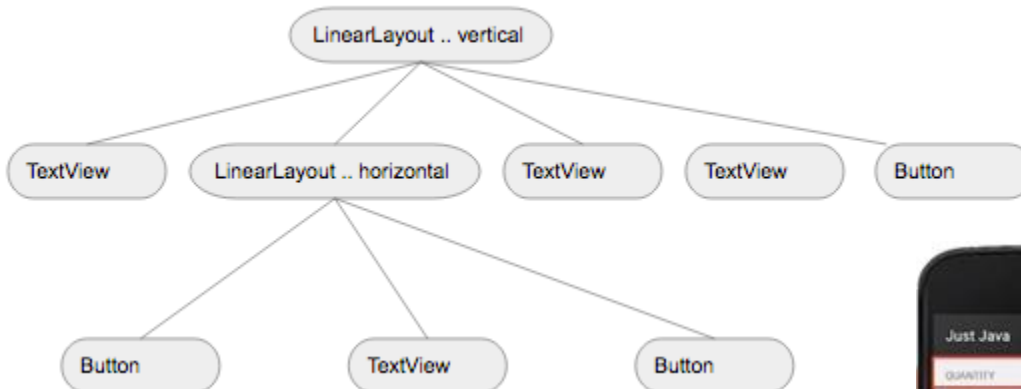
Л4. Activity. Этап 2. Nested ViewGroups

Nested ViewGroups – це вкладені групи, які знаходяться всередині інших груп. Приклад *Vertical LinearLayout*, який складається з елементів *Horizontal LinearLayout*.

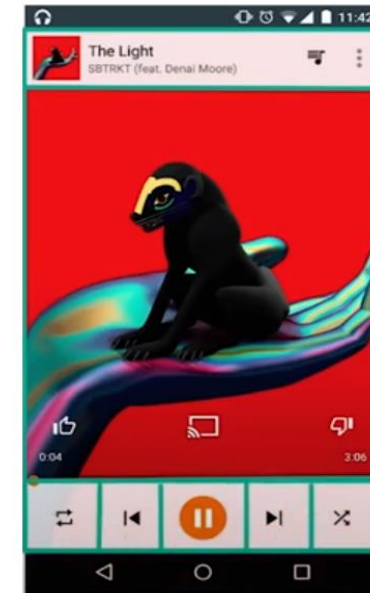
| | | |
|---|-----------|---|
|  | Red Sox | 4 |
|  | Cardinals | 2 |
| Final, October 27 - World Series | | |
| Recap & highlights | | |
| Box score | | |

| Stocks today | | | |
|----------------------------|----------|-------|--------|
| Updated: May 8, 5:48pm EST | | | |
| GOOG | 1,130.18 | 16.00 | 1.44% |
| CLX | 71.76 | -0.46 | -0.64% |
| F | 9.09 | 0.17 | 1.91% |
| VZ | 44.46 | -0.16 | -0.36% |
| TMUS | 31.86 | -0.17 | -0.53% |

[Show more](#) Disclaimer



Vertical Linear Layout (vertical orientation)



Relative Layout

Relative Layout

Linear Layout (Horizontal orientation)



Vertical Linear Layout Horizontal Linear Layout

Л4. Activity. Этап 2. Nested ViewGroups

```
<LinearLayout 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"  
android:orientation="vertical"  
android:padding="16dp"  
tools:context=".MainActivity">
```

```
<TextView
```

```
android:id="@+id/Quan"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginBottom="10dp"  
android:text="Quantity"  
android:textAllCaps="true"/>
```

```
<LinearLayout
```

```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:orientation="horizontal">
```

```
<Button
```

```
android:id="@+id/minus_botton_view"  
android:layout_width="48dp"  
android:layout_height="48dp"  
android:layout_marginBottom="10dp"  
android:onClick="countMinus"  
android:text="-" />
```

Файл activity_main.xml

```
<TextView
```

```
android:id="@+id/quantity_text_view"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginBottom="10dp"  
android:layout_marginLeft="8dp"  
android:layout_marginRight="8dp"  
android:text="0"  
android:textColor="#000000"  
android:textSize="16sp" />
```

```
<Button
```

```
android:id="@+id/plus_botton_view"  
android:layout_width="48dp"  
android:layout_height="48dp"  
android:layout_marginBottom="10dp"  
android:onClick="countPlus"  
android:text="+" />
```

```
</LinearLayout>
```

```
<TextView
```

```
android:id="@+id/Price"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginBottom="10dp"  
android:text="PRICE" />
```

```
android:textColor="#000000"  
android:textSize="16dp" />
```

```
<Button
```

```
android:id="@+id/order_botton_view"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginBottom="10dp"  
android:onClick="submitOrder"  
android:text="ORDER" />
```

```
LinearLayout>
```


Л4. Activity. Етап 2. Nested ViewGroups

```
package com.example.justjava;
```

```
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
import java.util.Locale;
```

// This app displays an order form to order coffee.

```
public class MainActivity extends AppCompatActivity {
    int numberOfCofees = 0;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
```

// This method is called when the order button is clicked.

```
public void submitOrder(View view) {
    displayPrice(numberOfCofees * 5);
}
```

```
public void countPlus(View view) {
    numberOfCofees++;
    display(numberOfCofees);
}
```

MainActivity.java (без змін)

```
public void countMinus(View view) {
    if(numberOfCofees>1) numberOfCofees--;
    display(numberOfCofees);
}
```

//This method displays the given quantity value on the screen.

```
private void display(int number) {
    TextView quantityTextView = (TextView) findViewById(R.id.quantity_text_view);
    quantityTextView.setText("" + number);
}
```

//This method displays the given price on the screen.

```
private void displayPrice(int number) {
    TextView priceTextView = (TextView) findViewById(R.id.price_text_view);
    priceTextView.setText(java.text.NumberFormat.getCurrencyInstance(new
    Locale("ua", "UA")).format(number));
}
```


Л4. Activity. Створюємо інтерактивний додаток. Етап 1

Крок 17. Допрацюємо додаток, щоб появился надпис *Total, Thank you*

1) Додаємо метод **displayMessage ()**

```
private void displayMessage(String message)
{
    TextView priceTextView = (TextView)findViewById(R.id.price_text_view);
    priceTextView.setText(message);
}
```

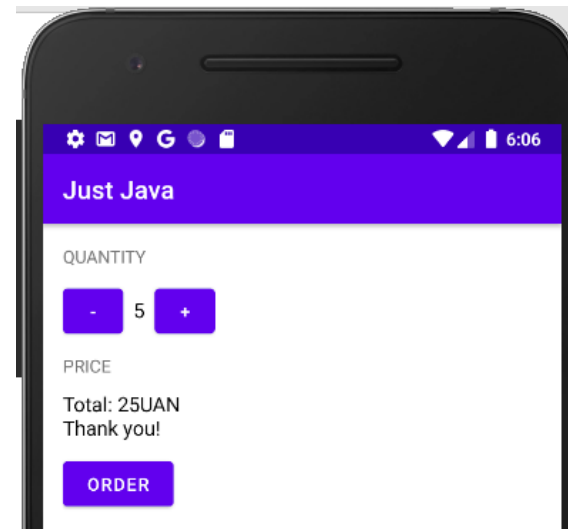
Змінна має починатися з маленької букви, якщо в назві декілька слів, то використовуємо стиль **camelCase**. При формуванні рядкової змінної, якщо всередині є лапки, то отримуємо повідомлення про помилку

```
String priceMessage = "She said "1 dollar"";
```

```
String priceMessage = "She said \" 1 dollar \"";
```

2) Вносимо зміни в метод **submitOrder()**

```
public void submitOrder(View view)
{
    String message = "Total: " + numberOfCofees * 5 + "UAN" + "\nThank you!";
    displayMessage (message);
}
```



Escape Sequences

| Escape Sequence | Description |
|-----------------|--|
| \t | Insert a tab in the text at this point. |
| \b | Insert a backspace in the text at this point. |
| \n | Insert a newline in the text at this point. |
| \r | Insert a carriage return in the text at this point. |
| \f | Insert a form feed in the text at this point. |
| \' | Insert a single quote character in the text at this point. |
| \" | Insert a double quote character in the text at this point. |
| \\ | Insert a backslash character in the text at this point. |