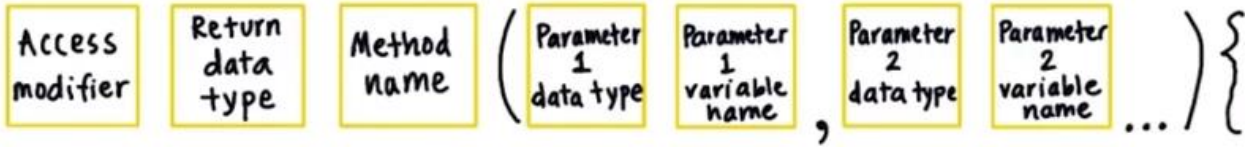
Л5.Методи

Task 1

У JAVA визначення метода ще називається сігнатура метода. Сігнатура метода – це тільки його назва та тип параметрів. Більш детально з особливостями застосування методів можна ознайомитись в документації по Java. Приклад оголошення методу:

```
private int calculatePrice(int quantity) {}
```




```
//Here is an example of a typical method declaration:
public double calculateAnswer(double wingSpan, int numberOfEngines, double length, double grossTons)
{
//do the calculation here
}
```

Task2

Параметри - це те, що вказується при створенні методу, їх приймає метод та обробляти всередині. При створенні методу оголошуємо та перераховуємо які параметри він буде приймати при виклику.

Аргументи - це те, що передається методу при його виклику, вхідні дані, які він прийме на обробку. При виклику методу передаємо йому фактичні значення – аргументи.

Виконання методу завершується:

- коли виконали усі рядки в методі;
- коли досягнули оперетора повернення *return*, який є першим;
- метод завершиться якщо викликається виняток або помилка.

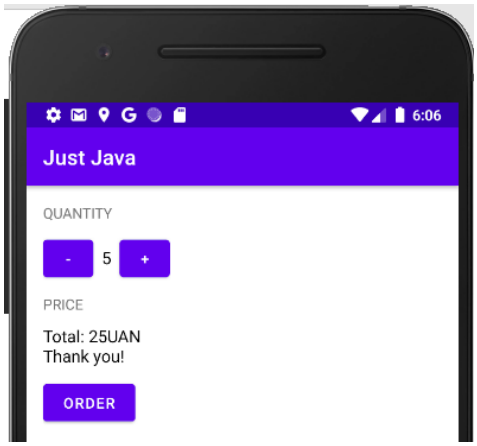
У коді може зустрітись `return.` Це вказує, що метод завершується і будь-який наступний рядок у методі не буде виконуватись. Може також зустрітись код – `return returnValue.` Це є повернення вхідного параметру з методу.

```
// a method for computing the area
public int getArea() {
    return widht*height;
}
```

Тип даних, що повертає даний метод `int widht*height.`

Task3

Продовжуємо роботу над додатком JustJava (лекція 4).



1. У файлі activity_main.xml:
   - змінюємо текст PRICE на ORDER SUMMARY;
   - TextView зі значення ціни задаємо ідентифікатор @+id/order_summary_text_view.

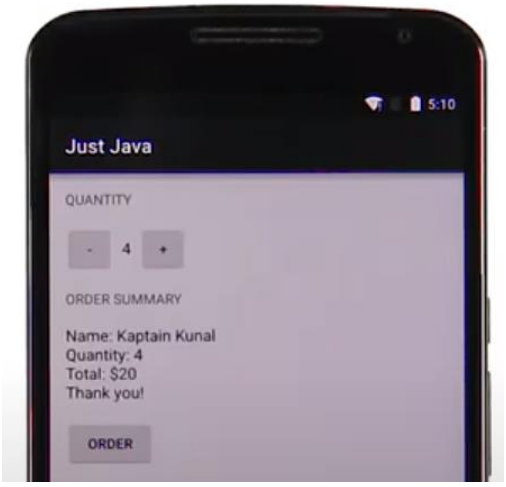
2. У методі displayMessage файлу MainActivity.java:
   - змінюємо назву змінної priceTextView на orderSummaryTextView;
   - змінюємо R.id.price_text_view на R.id.order_summary_text_view.

3. У файлі MainActivity.java:
   - видалити або закоментувати (**ctrl** + / (рядок) або **ctrl** + **shift** + / (фрагмент коду)) метод displayPrice;
   - внести зміни або додати методи submitOrder, calculatePrice, createOrderSummary.

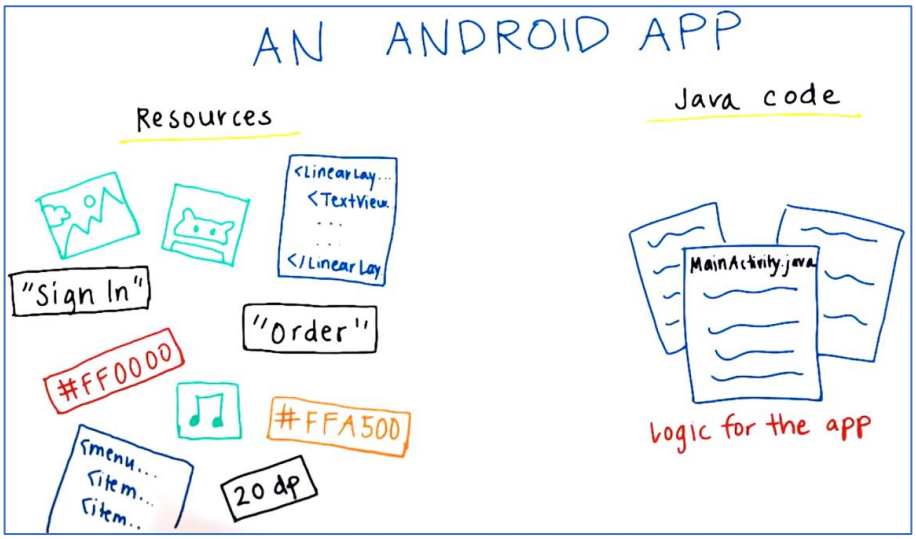
```
int numberOfCofees = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void submitOrder(View view) {
        int price = calculatePrice();
        displayMessage(createOrderSummary(price));
    }
   private int calculatePrice() {
        int basePrice = 5;
        return numberOfCofees * basePrice;
    }
   private String createOrderSummary(int price) {
        String priceMessage = "Name: Mallyness\n";
        priceMessage += "Quantity: " + numberOfCofees  + " \n";
```

```
        priceMessage += "Total: " + price + "$ \n";
        priceMessage += "Thank you!";
        return priceMessage;
    }
   private void displayQuantity(int number) {
        TextView quantityTextView = (TextView)
        findViewById(R.id.quantity_text_view);
        quantityTextView.setText("" + number);
    }
   public void countPlus(View view) {
        numberOfCofees++;
        display(numberOfCofees);
    }
   public void countMinus(View view) {
        if(numberOfCofees>1) numberOfCofees--;
        display(numberOfCofees);
    }
    private void display(int number) {
        TextView quantityTextView = (TextView)
        findViewById(R.id.quantity_text_view);
        quantityTextView.setText("" + number);
    }
    private void displayMessage(String message) {
        TextView orderSummaryTextView = (TextView)
        findViewById(R.id.order_summary_text_view);
        orderSummaryTextView.setText(message);
    }
}
```

Після виконання завдання маємо отримати вигляд



Ідентифікатори ресурсів. Об’єкти



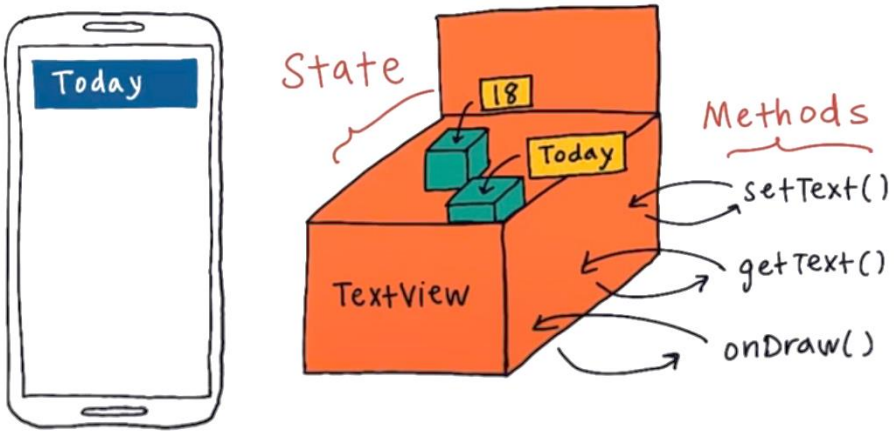
[Огляд різних ресурсів проектів]
[Підтримка різних екранів]

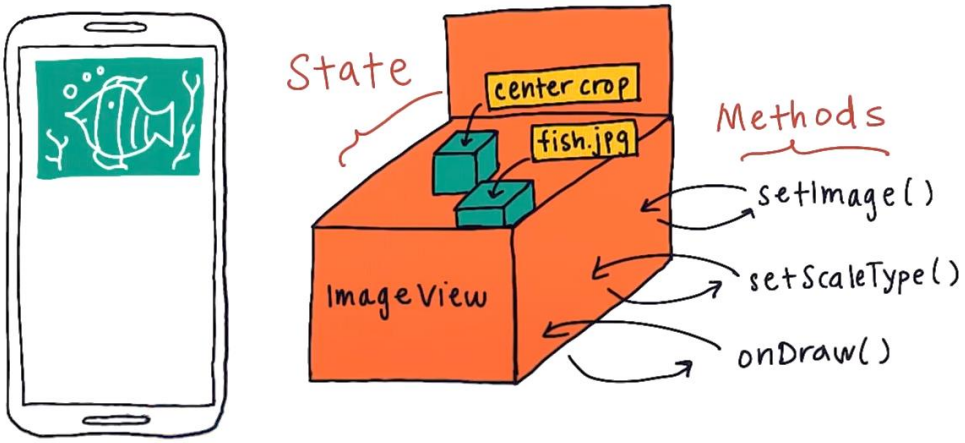
Ресурси Android (/res) можна поміщати у різні папки. Для звернення до графічних рисунків використовується формат *R.drawable*, для рядків *R.string.hello* де *R* це клас, *hello* це ім'я ресурсу. В коді вказуємо ресурси: *R.тип.имя_ресурса*, в розмітці вказуємо ресурси: *@тип/имя_ресурса*.

Приклад ресурсів та доступу до них

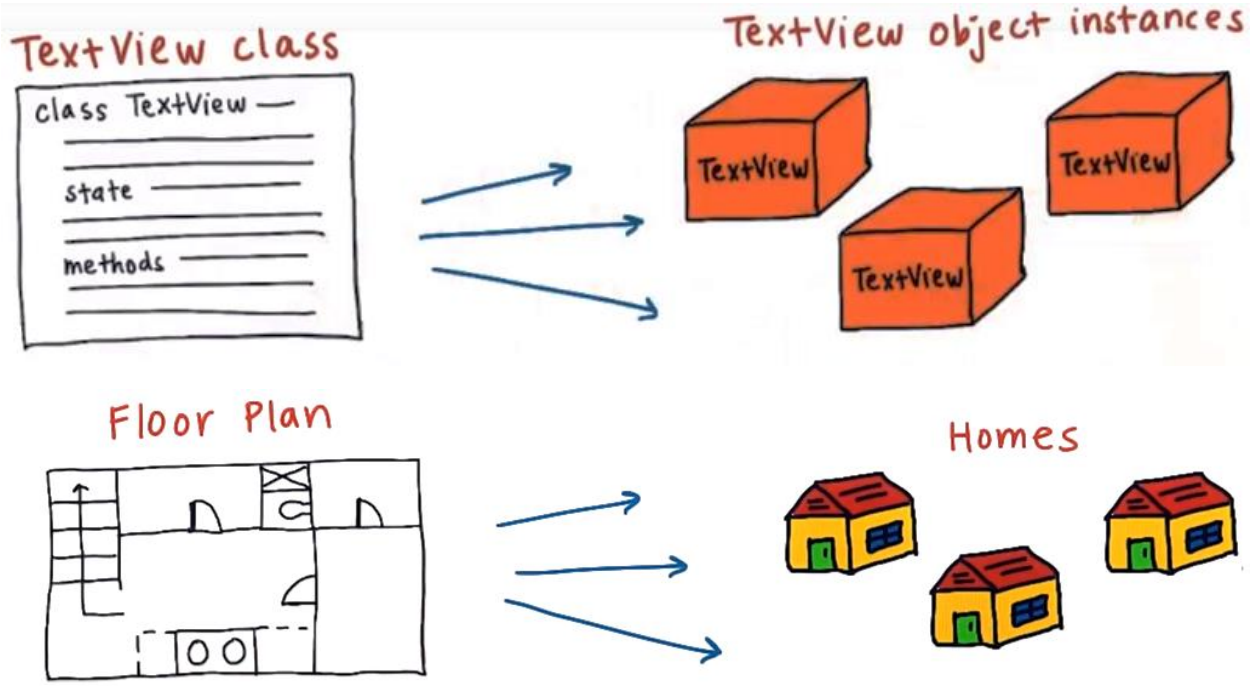
Resource Type	In Java code	In XML
Image	R.drawable.photo	@drawable/photo
String "Hello"	R.string.hello	@string/hello
Layout XML file	R.layout.activity_main	@layout/activity_main
ID	R.id.pice_text_view	@id/price_text_view
Color #FF0000	R.color.red	@color/red

Об'єкти та їх методи в Java





TextView це клас, який має поля стану та методи. Цей клас використовується для створення об’єктів (екземплярів) класу з різним змістом та виглядом



План дома не є самим домом. Кожний дом відрізняється але усі вони мають однаковий каркас (план).

Кілька корисних посилань:

- Детальніше про класи в Java
- Спрощений код TextView
- Спрощений код ImageView
- Повний код TextView
- Повний код ImageView

Щоб отримати доступ до вихідного коду, вбудованому в документацію будь-якого з класів Андроїд, встановіть це розширення для Google Chrome

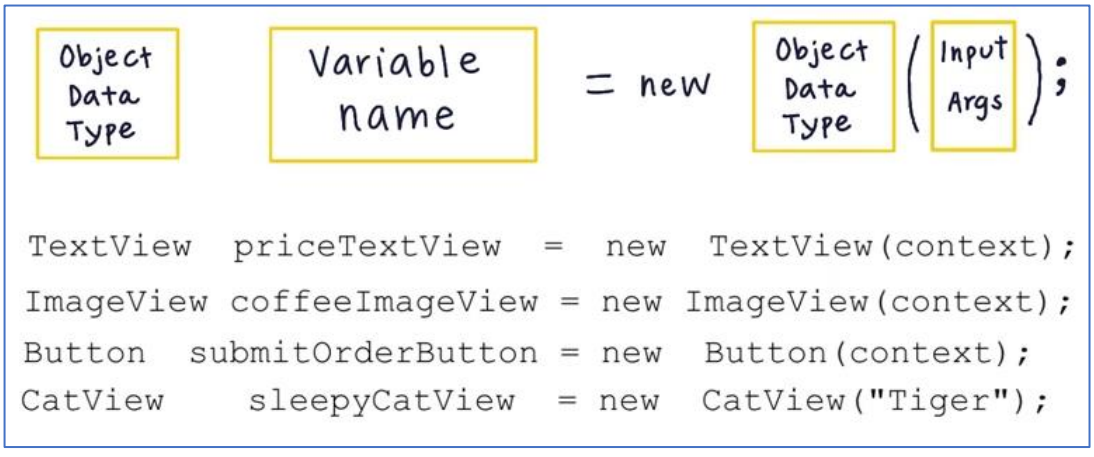
Клас в Java визначається ключовим словом class, назвою TextView, деяких параметрів та {}.

```
public class ImageView extends View {

    // Resource ID for the source image that should be displayed in the ImageView.
    private int mImageId;

    // Context of the app
    private Context mContext;

    /**
     * Constructs a new ImageView.
     */
    public ImageView(Context context) {
        mImageId = 0;
        mContext = context;
    }

    /**
     * Sets the source image in the ImageView.
     *
     * @param imageId is the resource ID of the image to be displayed.
     */
    public void setImage(int imageId) {
        mImageId = imageId;
    }
}
```

змінні починаються з m (Member Variable)

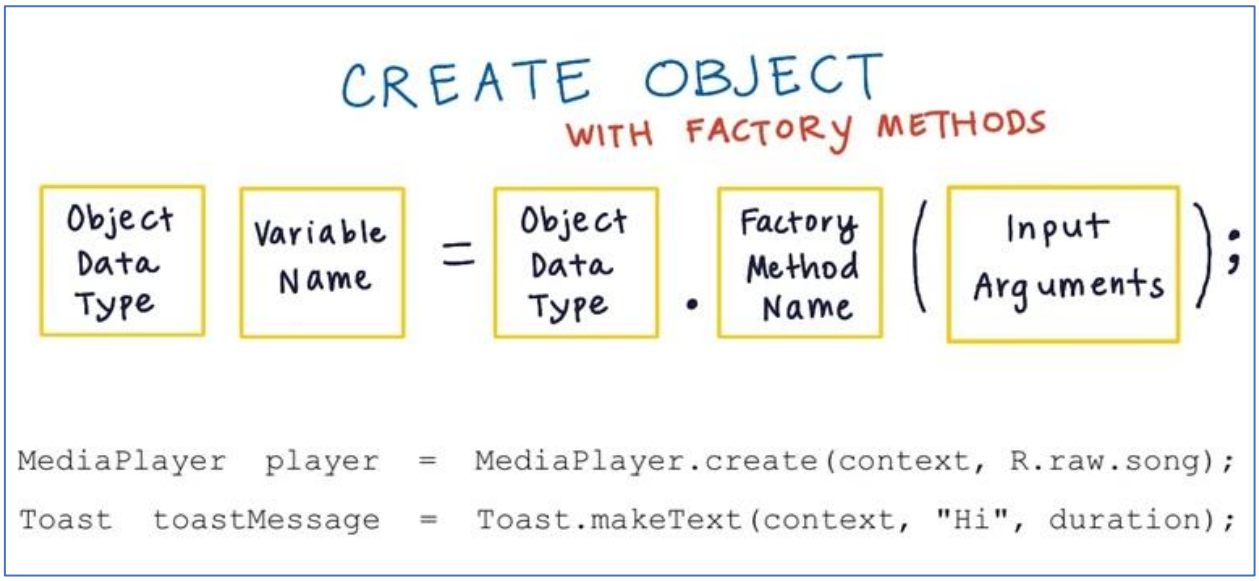
конструктор класу

Конструктори викликаються щоб створити екземпляр класу.



Contex об’єкт дозволяє отримати доступ до ресурсів та інших частин додатку.

Інколи може бути використаний такий алгоритм створення об’єкта

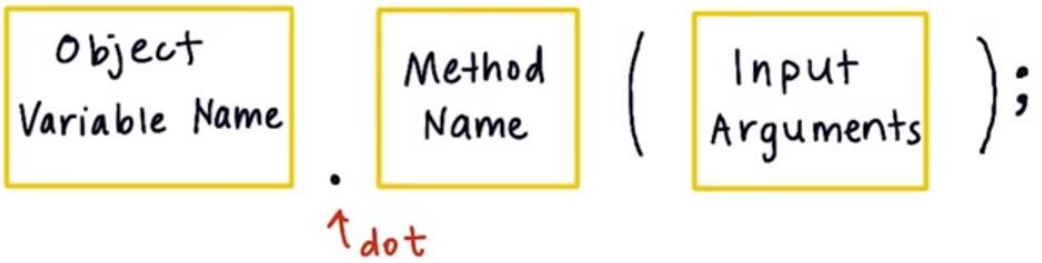


Task 4

Розглянемо ще раз метод `displayMessage`

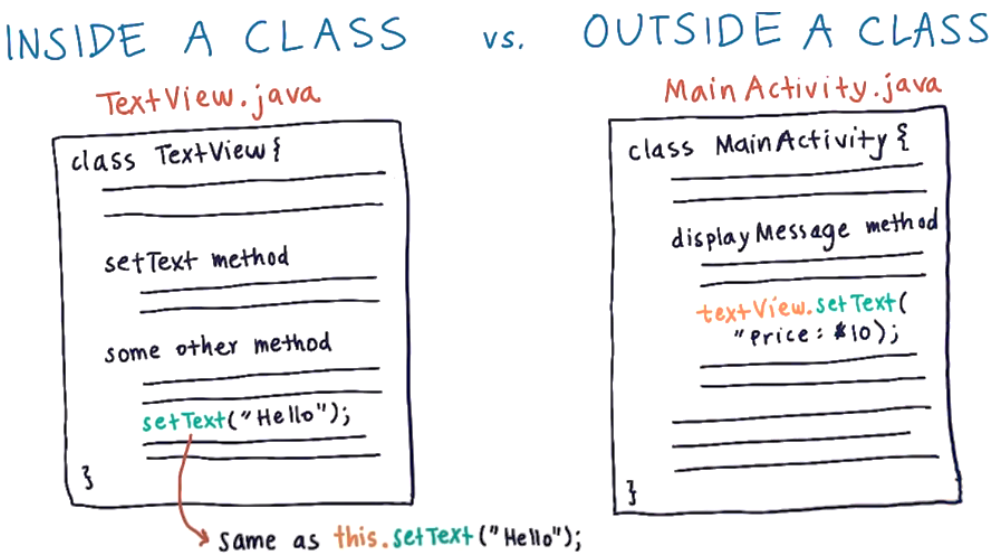
```
private void displayMessage(String message) {
    TextView orderSummaryTextView = (TextView) findViewById(R.id.order_summary_text_view);
    orderSummaryTextView.setText(message);
}
```

Він містить об'єкт TextView з ім'ям `orderSummaryTextView`. У наступному рядку викликаємо метод `setText` для даного об'єкту. Для запуску методу на об'єкті у нього буде такий формат




```
titleTextView.setText("News");
titleTextView.setTextSize(18);
warningTextView.setTextColor(Color.RED);
welcomeImageView.setImageResource(R.drawable.cloud);
```

Якщо знаходимся в середині класу і викликаємо метод `setText` з іншого методу даного класу то при виклику методу крапка не ставиться. Якщо викликаємо метод класу зовні з іншого місця, тоді вказується через точку. Доступ є тільки до public змінних та методів.



Inheriting Behavior (Успадковуємо поведінку)

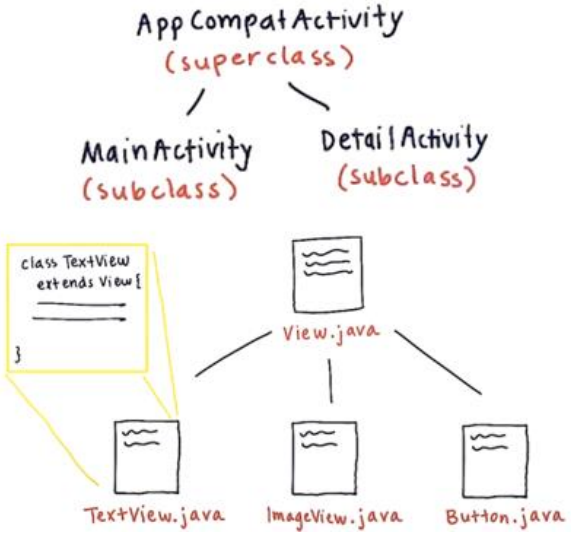
У попередньо розглянутому коді що означає `findViewById`?

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

Якщо переглянути повний код, то ніде не знайдемо метод `findViewById`. Код **MainActivity.java** не великий, але можна побачити такий рядок з кодом `extends AppCompatActivity`:

```
public class MainActivity extends AppCompatActivity {

    int numberOfCofees = 0;
```

Це значить, що **MainActivity** це розширення класу **AppCompatActivity** і відповідно MainActivity успадковуємо (отримуємо доступ) до методів AppCompatActivity.

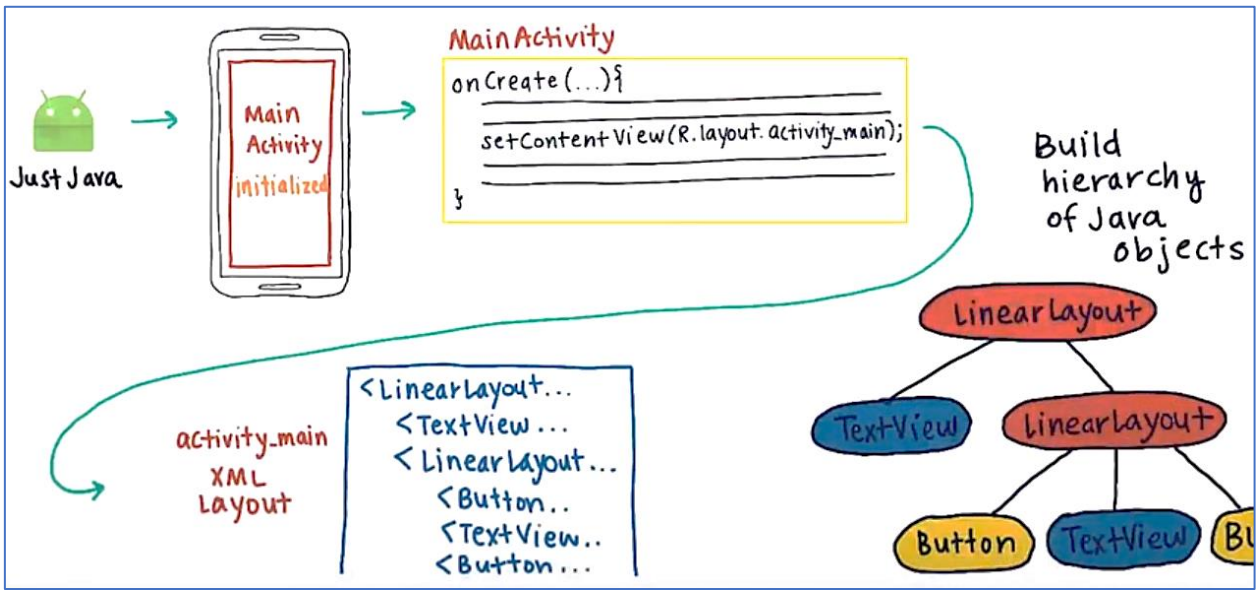


Класи TextView, ImageView, Button успадковують усі властивості та методи класу View але в них знаходяться методи та поля, що притаманні тільки ним.

Діаграма ієрархії класів

Приведення типів даних

Розглянемо як працює додаток JustJava.



Якщо натиснути на іконку додатку на пристрої відкривається основна функція **MainActivity** – **onCreate**. У методі викликається метод **setContentView** з макетом ресурсу **activity.main**. Так як хочемо мати більше 1 елементу для додатку, то вводимо макет ресурсу та прив'язуємо до **xml** файлу. Функція знаходить файл макету **xml** та зчитує кожний рядок та будує повну ієрархію об'єктів java. Під час створенная кожного з елементів викликається конструктор **new TextView** або **new LinearLayout**, далі передаються всі дані атрибути в объект java. В XML файлі для ідентифікації об'єктів їм присвоюємо ID. В класі **AppCompatActivity** є метод **findViewById (int id)**, який знаходить елемент за його ID.

Результат отриманий з findViewById (R.id.id_name) повертає View згідно документації. Однак в правій частині створюється елемент TextView, значить потрібно View привести до типу TextView. Це робиться через дужки з TextView

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

Task5

Розглянемо код методу

```
TextView orderSummaryTextView = (TextView) findViewById(R.id.order_summary_text_view);
orderSummaryTextView.setText(message);
```

Як використовувати метод setText для зміни View-елемента? Це робиться в два етапи:

Крок 1: Отримання об'єкта **View** за допомогою його ідентифікатора. У першому рядку коду отримуємо **TextView** і зберігаємо його в змінну під назвою **orderSummaryTextView** (текстове поле замовлення). Щоб отримати **View**, використовуємо метод **findViewById** (отримати view за ідентифікатором), це метод з класу **Activity**. У нього передаємо один аргумент - ідентифікатор **view**, цей аргумент надаємо в вигляді **R.id.IDOFVIEW**. В даному випадку ідентифікатор View заданий в XML **- order_summary_text_view**. Далі відбувається приведення об'єктів, цим займається **(TextView)**. Тобто повертається значення методу **findViewById** повинно бути типу **TextView**, а не просто View.

Крок 2: Виклик методу на об'єкті **View**. Оскільки викликаємо метод на об'єкті, потрібно використовувати виклик через точку. Рядок **orderSummaryTextView.setText (message)** - все одно що інструкція візьми об'єкт orderSummaryTextView, якому можна задати текст з рядка, який передали в метод (в даному випадку передали змінну типу String під назвою message).

Методи, як **setText** (встановити текст) і **setImageResource** (встановити картинку) називають сетерами. Вони встановлюють одне зі значень View (наприклад, зберігається в ньому текст або картинку). Вони традиційно починаються зі слова set (встановити).

Є ще така категорія методів, як геттери. Їх єдине призначення - отримати одне зі значень View, наприклад, текст, який на даний момент встановлено. Вони традиційно починаються зі слова get(отримати).