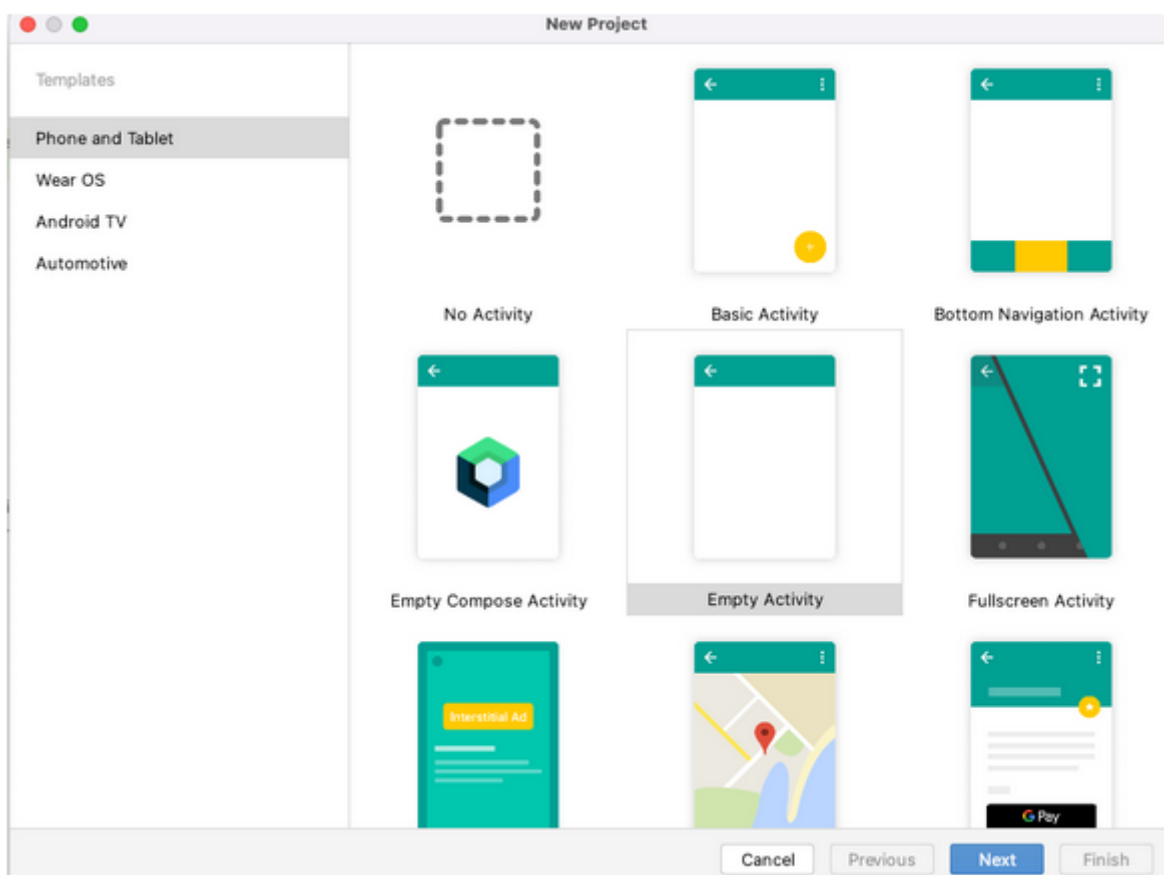


Практическая работа № 1 “Создание проекта. Верстка загрузочного экрана по макету”

в папке Видео к работам содержатся пояснения к практической работе

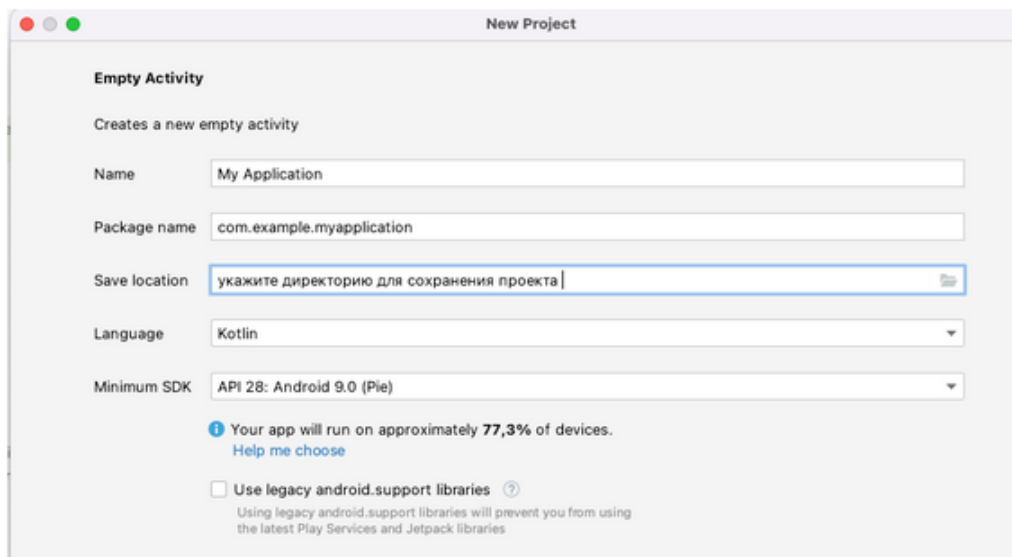
Запустите Android Studio и создайте новый проект, выполнив следующую последовательность действий (обращаю ваше внимание, если вы работаете в версии Flamingo - вам следует выбирать не просто Empty Activity, а Empty View Activity):

Для создания нового проекта, нажмите File | New | New Project.... Появится диалоговое окно мастера.



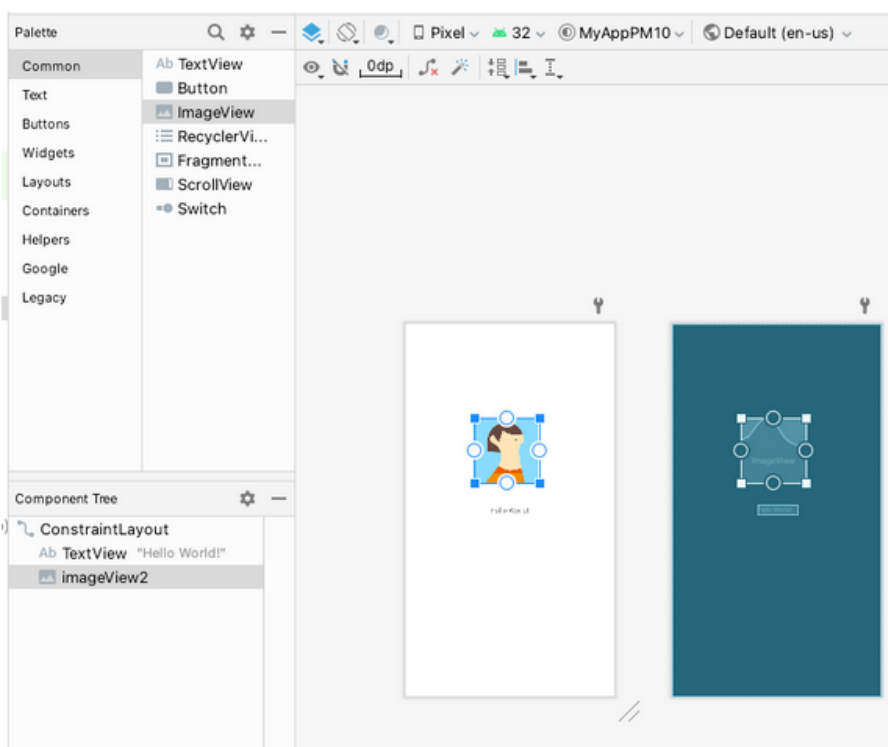
В категории Template (шаблоны) выбираете Phone and Tablet (телефоны и планшеты) тип проекта Empty Activity (пустой экран) - тип экрана нашего будущего приложения - и нажимаем Next.

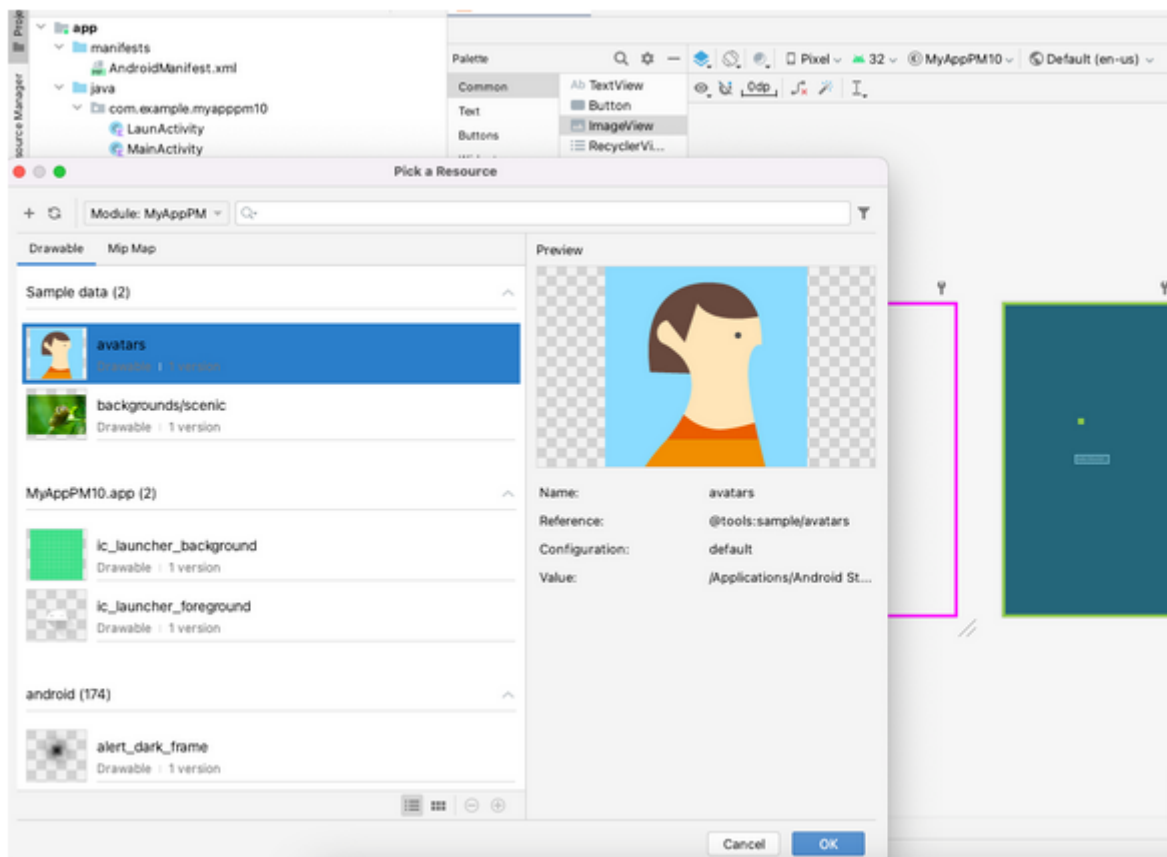
Заполните информацию о проекте и нажмите Finish (name - название приложения, save location - директория сохранения проекта, language - язык разработки (всегда устанавливаете Kotlin) minimum SDK - выбираете **API30**):



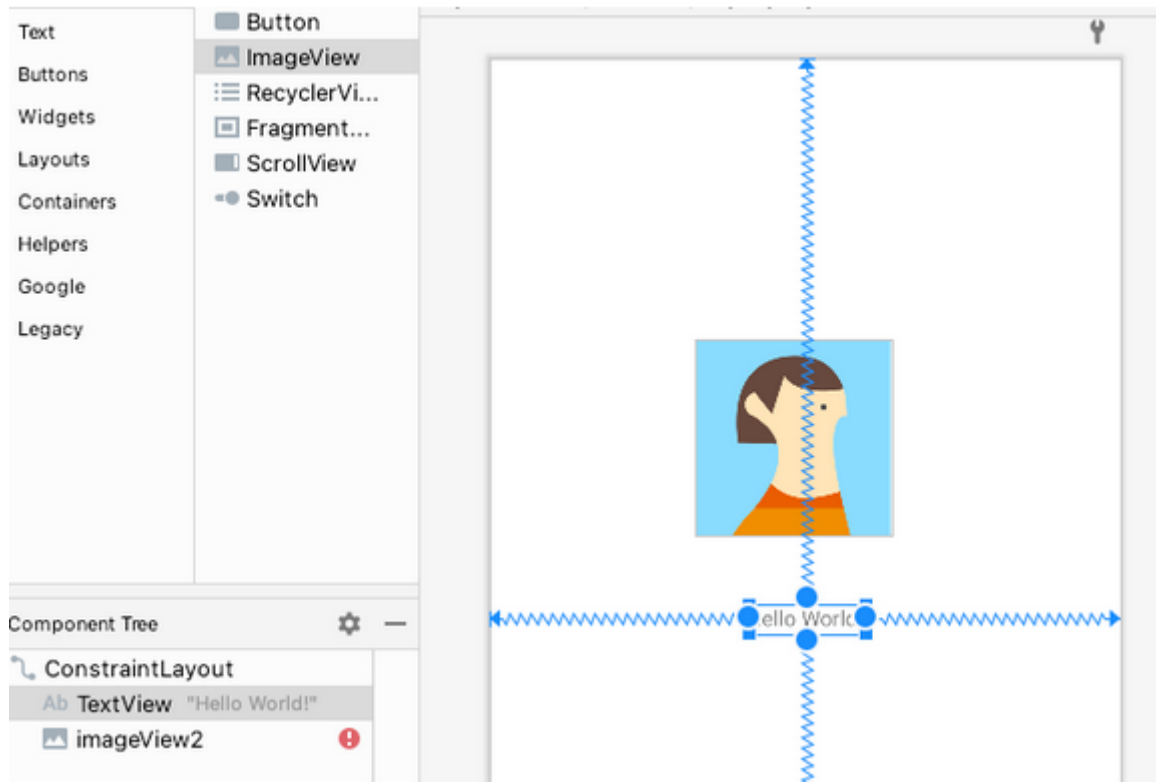
Дождитесь полной сборки проекта и приступите к верстке первого загрузочного экрана. На экране загрузки будет отображаться изображение главного героя вашего будущего приложения. Для размещения изображения на экране откройте файл разметки `activity_main.xml`, если он у вас закрыт. Убедитесь, что у вас установлен режим работы с файлом Design.

На панели инструментов Palette в группе Common найдите элемент `ImageView` (контейнер, используется для отображения изображений, вставки картинок), удерживая левую кнопку мыши, перетащите его на экран. У вас появится диалоговое окно `Pick a Resource` (укажите ресурс) с просьбой выбрать изображение. Выбирайте любое, позже вы его замените. Нажмите ОК.

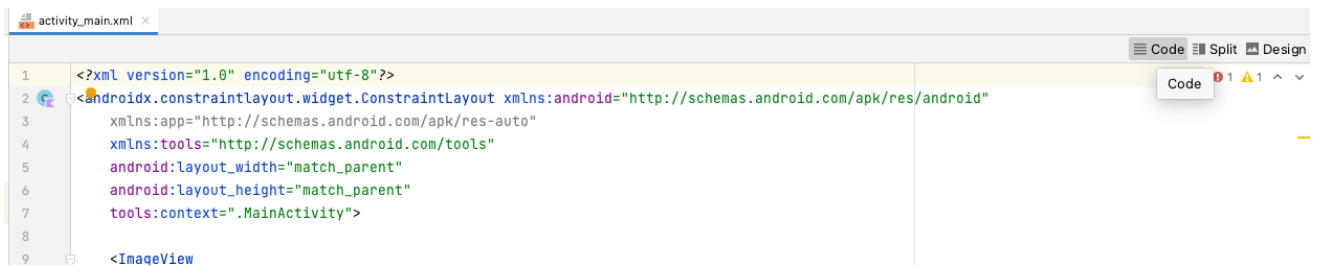




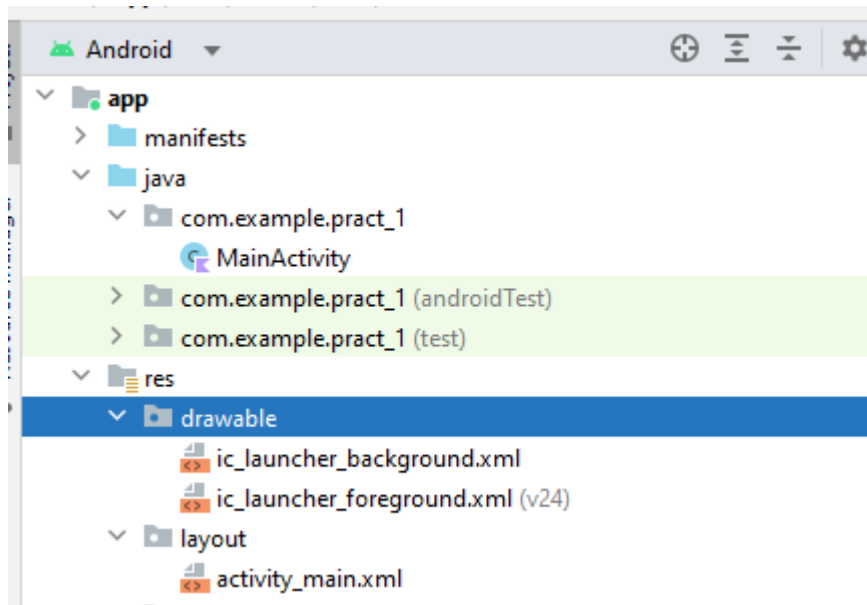
Текстовое поле вам не понадобится, выделите его и удалите:



Переключитесь в режим редактирования кода разметки, выбрав в правом верхнем углу вкладку Code:

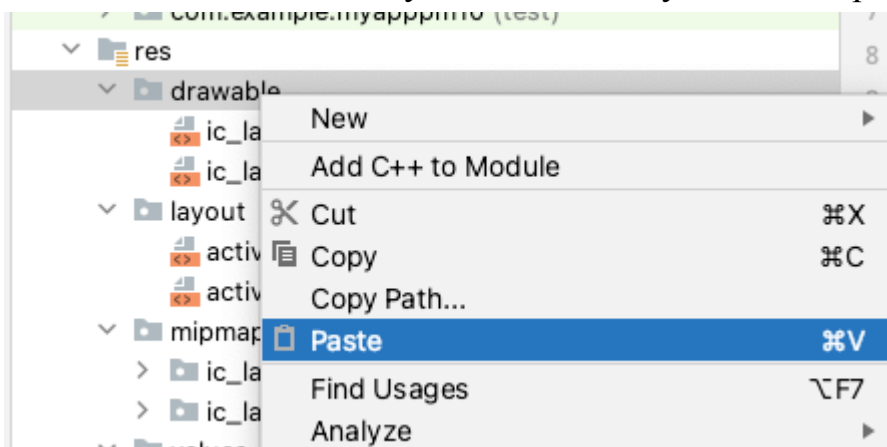


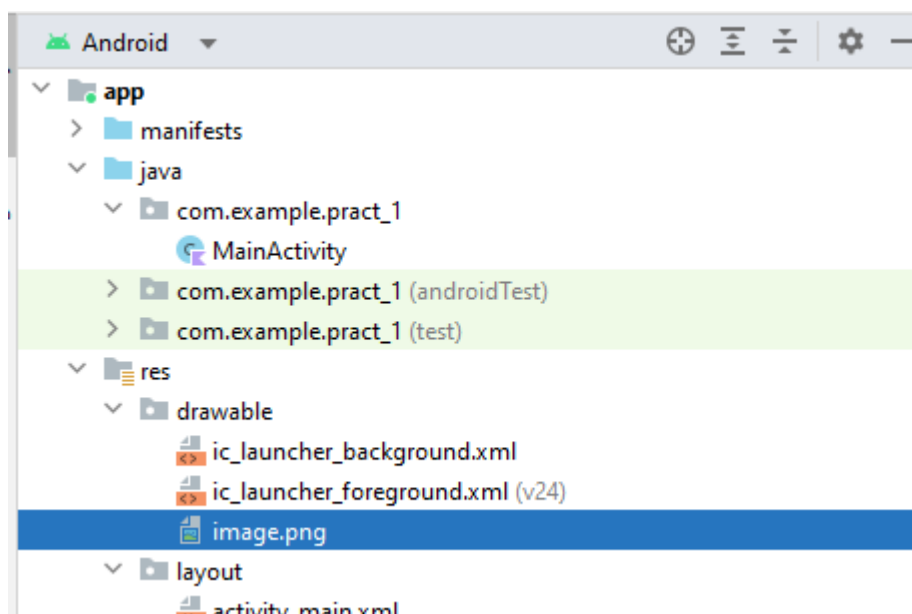
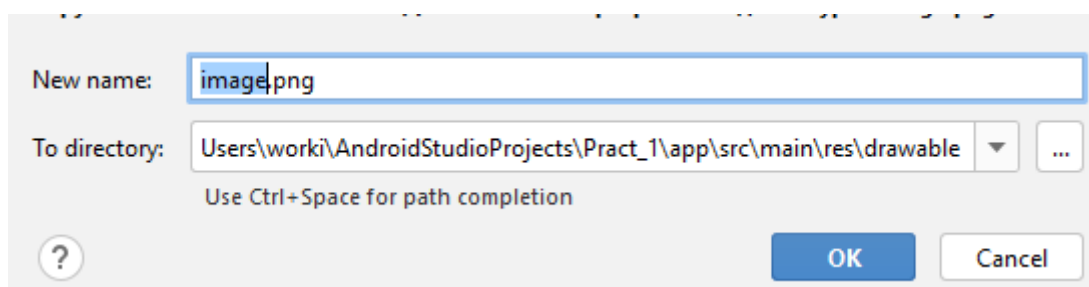
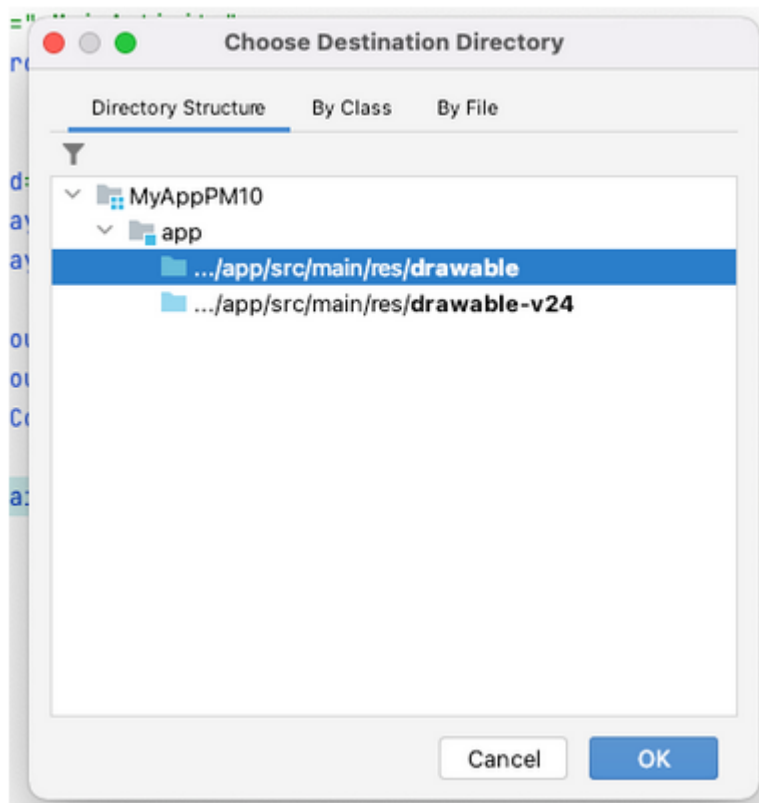
Все используемые в проекте графические изображения должны быть размещены в папке drawable



В папке Ресурсы будут отображаться все необходимые графические материалы к практическим работам. Скачайте изображение загрузка, обязательно переименуйте его в image.

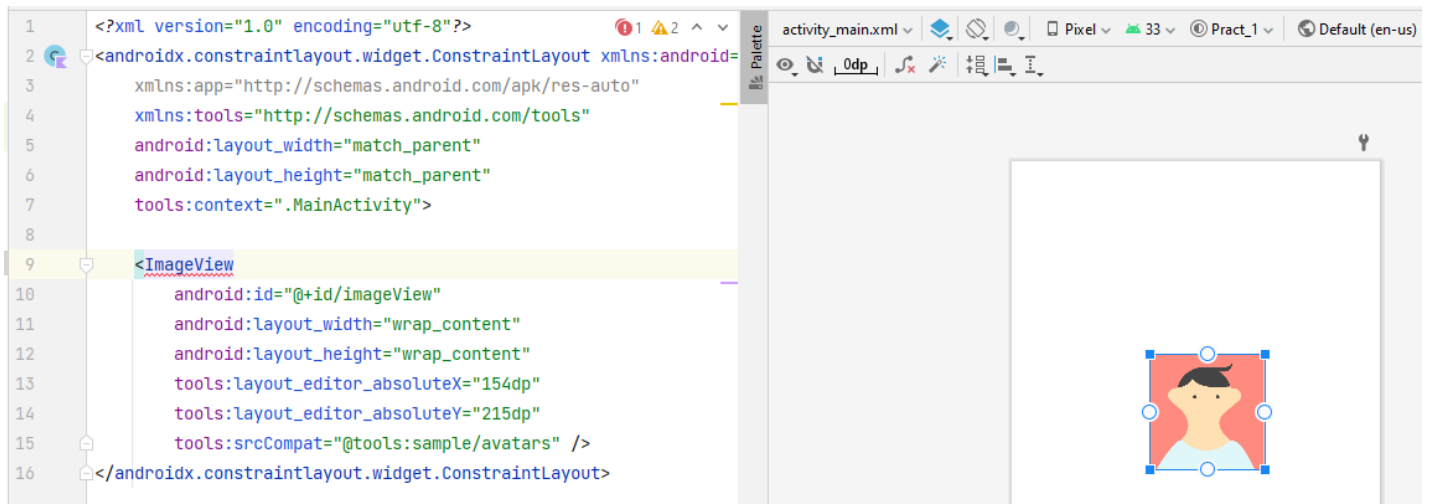
Скопируйте (ctrl+C) картинку image.png в буфер, затем щёлкните правой кнопкой мыши по папке drawable (res/drawable), выберите команду "Вставить" (Paste). В появившихся диалоговых окнах установите следующие настройки и нажмите **ОК**:





Запомните эту последовательность действий, все растровые изображения перед использованием необходимо скопировать в папку drawable.

Замените изображение, которое отображается в ImageView, на загруженную картинку.



Удалите строку кода `tools:srcCompat="@tools:sample/avatars"`

На месте удаленной строки напечатайте

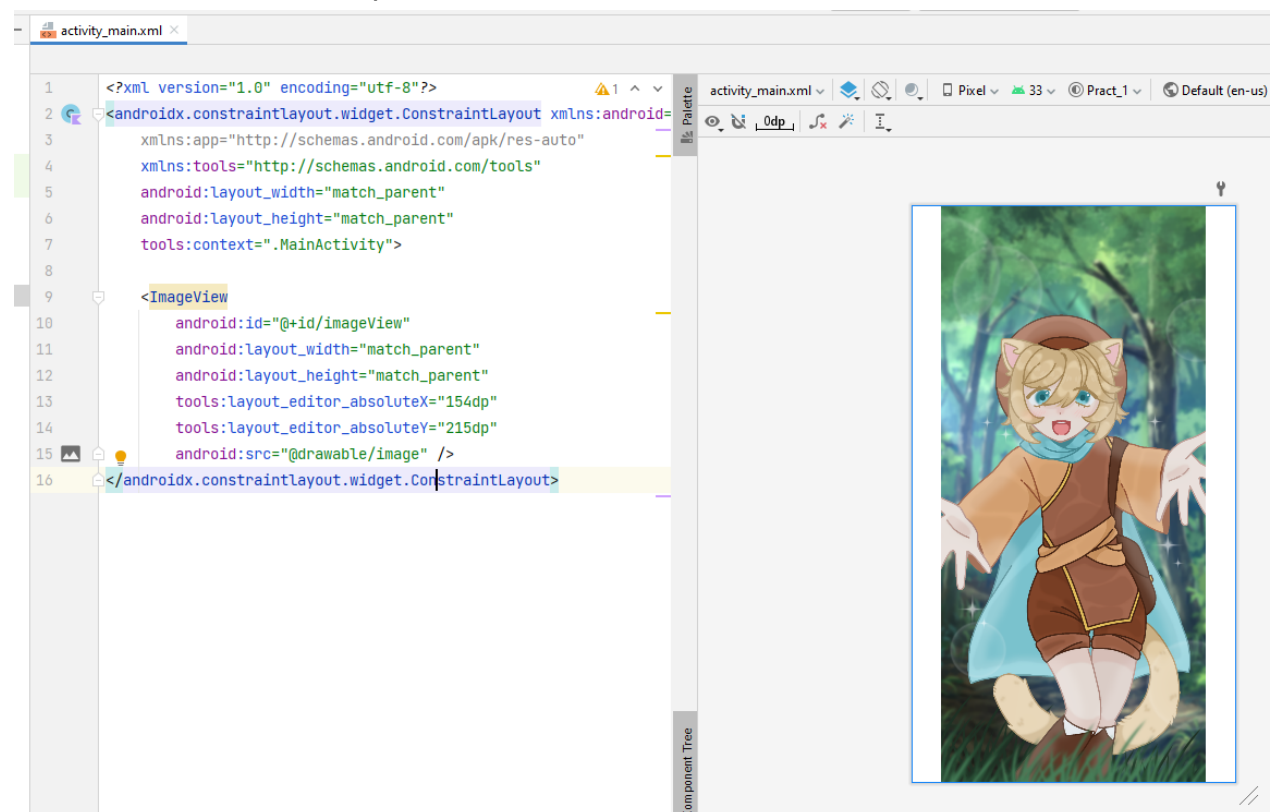
`android:src="@drawable/image"` – параметр `src` отвечает за то, какое изображение будет отображаться внутри `ImageView`

Изображение должно занимать весь экран, для параметров отвечающих за высоту и ширину `ImageView` установите значение `match_parent`.

`android:layout_width="match_parent"` – установить ширину по ширине родительского объекта

`android:layout_height="match_parent"` – установить высоту по высоте родительского объекта

На данном этапе ваш экран и код должны выглядеть так:



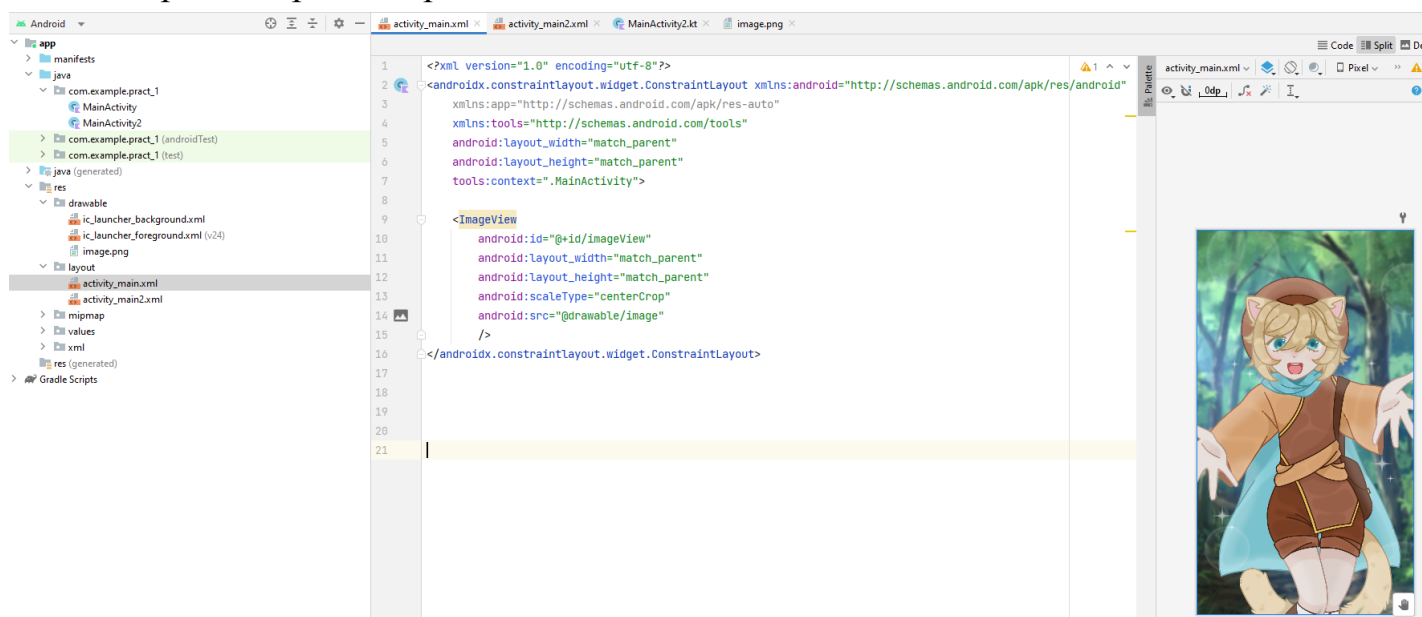
Настройте масштабирование изображения внутри контейнера `ImageView`, добавив параметр

`android:scaleType="centerCrop"` – настройка масштабирования изображения внутри `ImageView`

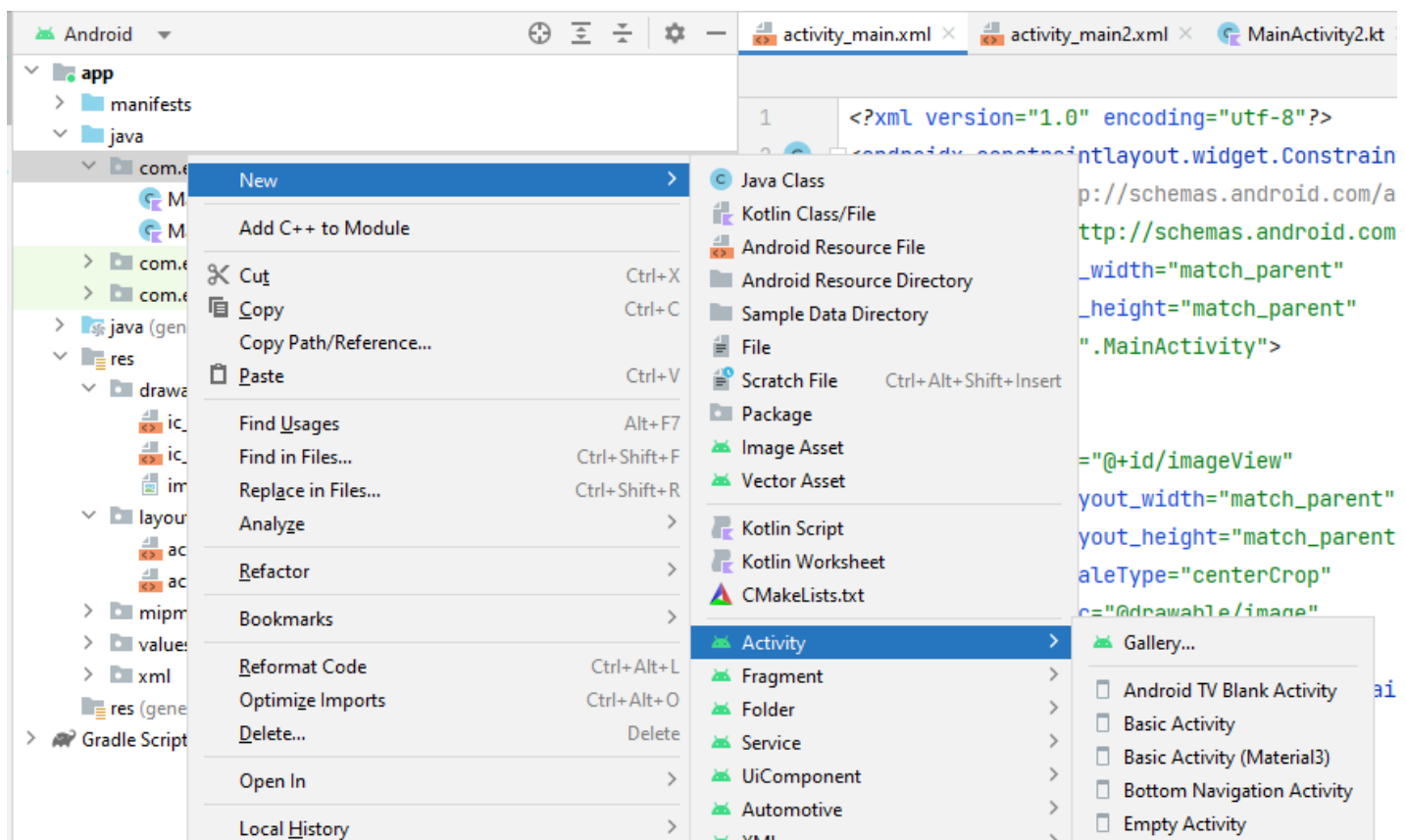
Более подробную информацию можно узнать перейдя по ссылкам:
<https://developer.android.com/reference/android/widget/ImageView.ScaleType>
<https://thoughtbot.com/blog/android-imageview-scaletype-a-visual-guide>

Попробуйте установить разные значения для параметра масштабирования и посмотрите как будет меняться отображение изображения на экране.

Верстка первого экрана закончена.



Добавьте в проект второй экран, создав новую Activity:

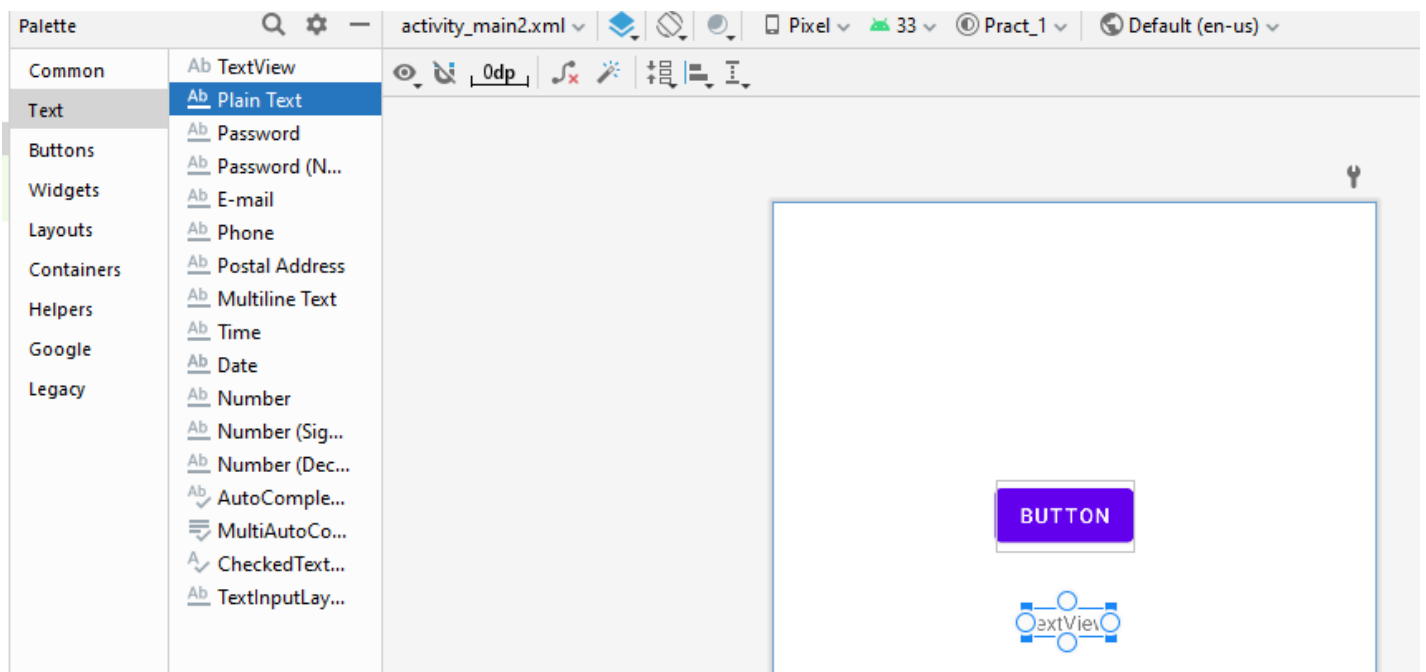
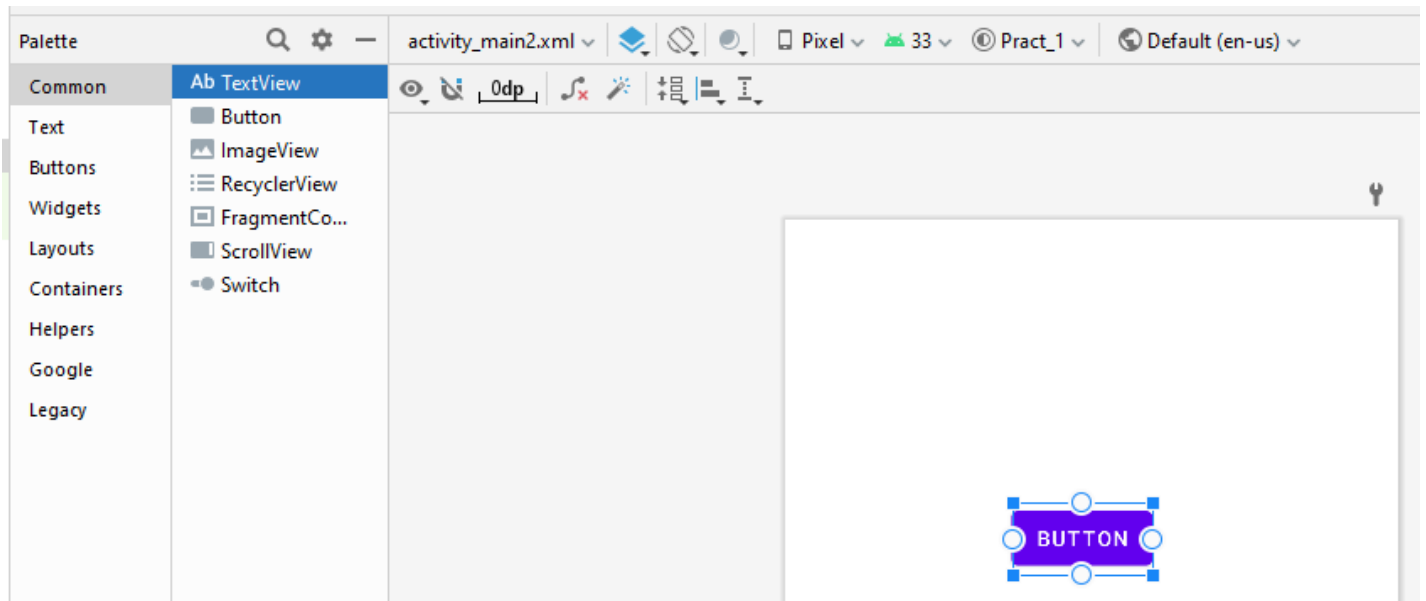
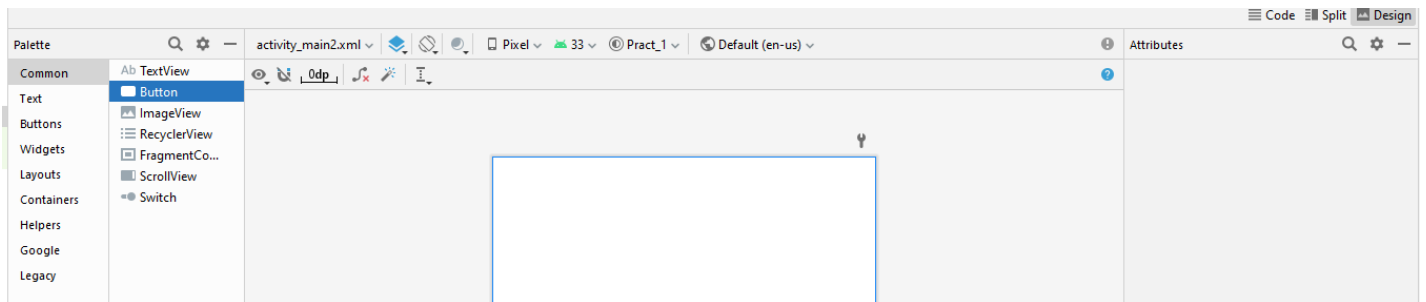


The 'New Android Activity' dialog box is shown. The 'Empty Activity' tab is selected. The 'Activity Name' field contains 'MainActivity2'. The 'Generate a Layout File' checkbox is checked. The 'Layout Name' field contains 'activity_main2'. The 'Launcher Activity' checkbox is unchecked. The 'Package name' field contains 'com.example.pract_1'. The 'Source Language' dropdown menu is set to 'Kotlin'.

На втором экране необходимо расположить следующие элементы интерфейса:

- **PlainText** - текстовое поле для ввода текста пользователем на экране приложения
- **TextView** - контейнер для отображения текста на экране
- **Button** - кнопка

В режиме Design, добавьте на экран указанные элементы пользовательского интерфейса:



A UI mockup within a light gray rectangular frame. At the top center is a purple rounded rectangle button with the word "BUTTON" in white capital letters. Below the button is the text "TextView" in a small, gray font. Further down is a text input field with the label "Name" in a bold, black font. The label "Name" is positioned to the left of a horizontal line, which is underlined in red.

Переключитесь в режим работы с кодом. Все элементы подчеркнуты красным.

```
activity_main.xml x activity_main2.xml x MainActivity2.kt x image.png x
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity2">
8
9     <Button
10         android:id="@+id/button"
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:text="Button"
14         tools:layout_editor_absoluteX="152dp"
15         tools:layout_editor_absoluteY="190dp" />
16
17     <TextView
18         android:id="@+id/textView"
19         android:layout_width="wrap_content"
20         android:layout_height="wrap_content"
21         android:text="TextView"
22         tools:layout_editor_absoluteX="172dp"
23         tools:layout_editor_absoluteY="277dp" />
24
25     <EditText
26         android:id="@+id/editTextTextPersonName"
27         android:layout_width="wrap_content"
28         android:layout_height="wrap_content"
29         android:ems="10"
30         android:inputType="textPersonName"
31         android:text="Name"
32         tools:layout_editor_absoluteX="115dp"
33         tools:layout_editor_absoluteY="342dp" />
34 </androidx.constraintlayout.widget.ConstraintLayout>
```

Это указывает на наличие ошибки в коде настройки элемента. У текстового поля, кнопки и текста отсутствуют привязки и программа просто не знает как эти элементы следует позиционировать на экране относительно других элементов интерфейса. Привязки должны быть установлены **всегда**.

Добавьте следующие строки кода для элемента TextView: привязка верхней части элемента к верхней части... привязка левой стороны привязка правой стороны...:

app:layout_constraintEnd_toEndOf="parent" - привязать правую сторону настраиваемого компонента к правой стороне родителя

app:layout_constraintStart_toStartOf="parent" - привязать левую сторону настраиваемого компонента к левой стороне родителя

app:layout_constraintTop_toTopOf="parent" - привязать верхнюю сторону настраиваемого компонента к верхней границе родителя

Более подробную информацию можно узнать перейдя по ссылкам:
<https://java-online.ru/android-layout.shtml>
<https://developer.android.com/reference/androidx/constraintlayout/widget/ConstraintLayout>
<https://metanit.com/java/android/3.8.php>

```
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity2">
8
9     <Button
10         android:id="@+id/button"
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:text="Button"
14         tools:layout_editor_absoluteX="152dp"
15         tools:layout_editor_absoluteY="190dp" />
16
17     <TextView
18         android:id="@+id/textView"
19         android:layout_width="wrap_content"
20         android:layout_height="wrap_content"
21         android:text="TextView"
22         app:layout_constraintTop_toTopOf="parent"
23         app:layout_constraintStart_toStartOf="parent"
24         app:layout_constraintEnd_toEndOf="parent"/>
```

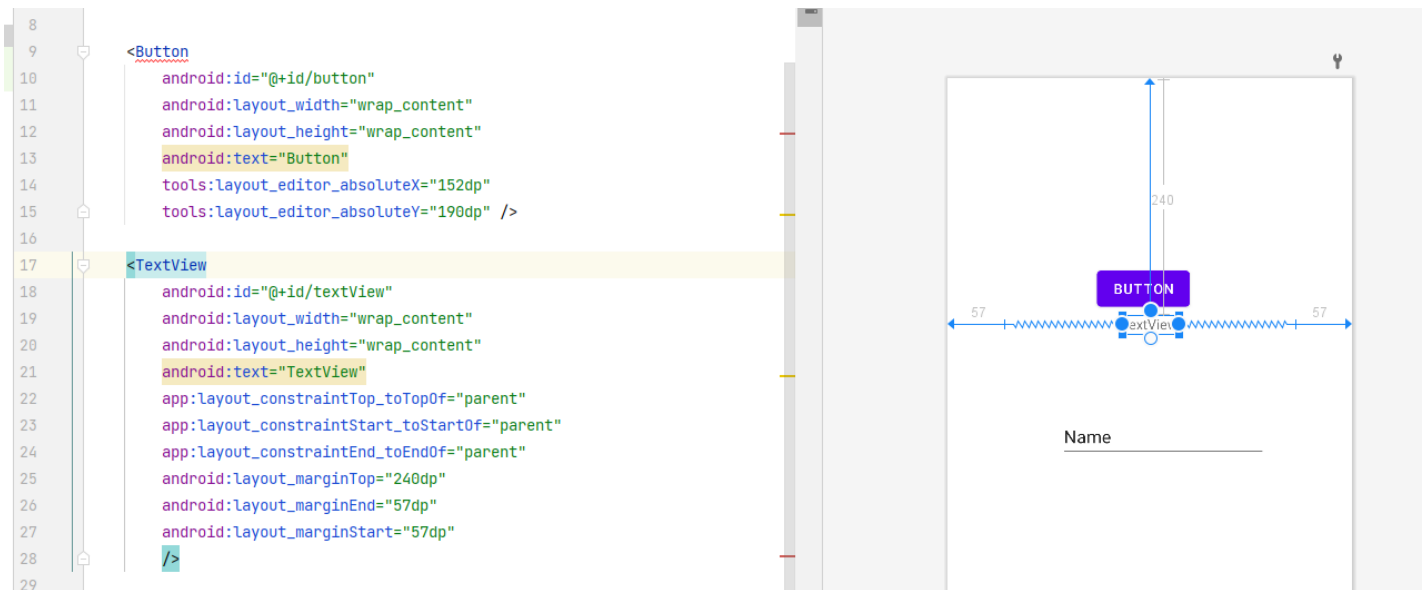
Вы “привязали” TextView к сторонам экрана (parent - привязка к родителю, к элементу, внутри которого расположен настраиваемый элемент).

Установите для TextView отступы - поля. Указанные значения будут измеряться относительно стороны того компонента, к которому был привязан настраиваемый элемент:

`android:layout_marginTop="240dp"` - установить верхнее поле (отступ сверху)

`android:layout_marginEnd="56dp"` - установить левое поле

`android:layout_marginStart="57dp"` - установить правое поле



Установите привязки для кнопки. Правую и левую стороны привяжите к родительскому объекту (parent), а верхнюю сторону кнопки привяжите к нижней стороне текстового поля.

Для того, чтобы привязать компонент к другому, не родительскому объекту, необходимо обратиться к этому объекту по его идентификатору. Значение идентификатора прописано в параметре

`android:id="@+id/editTextTextPersonName"` - уникальный идентификатор

`app:layout_constraintTop_toBottomOf="@+id/editTextTextPersonName"` -

привязать верхнюю сторону настраиваемого компонента к нижней стороне компонента, идентификатор которого указан

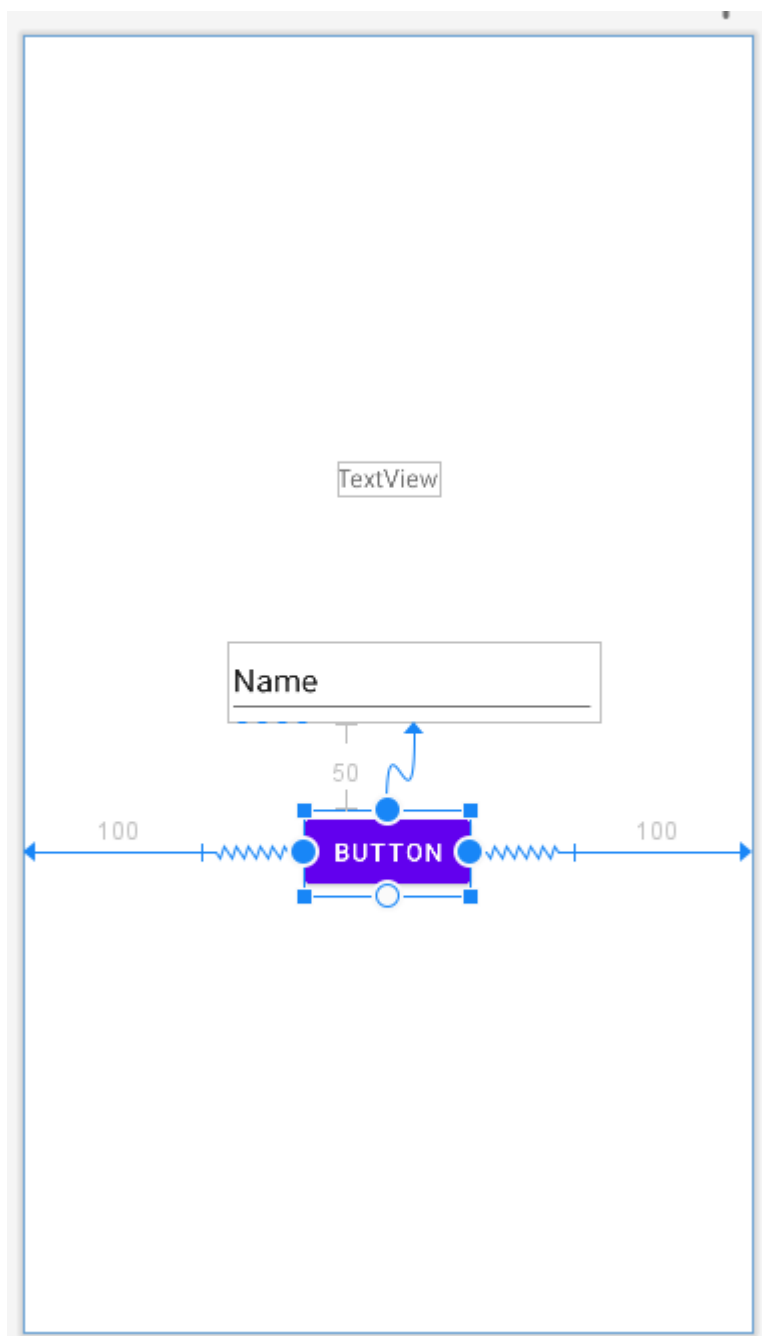


Самостоятельно установите для кнопки следующие значения полей:

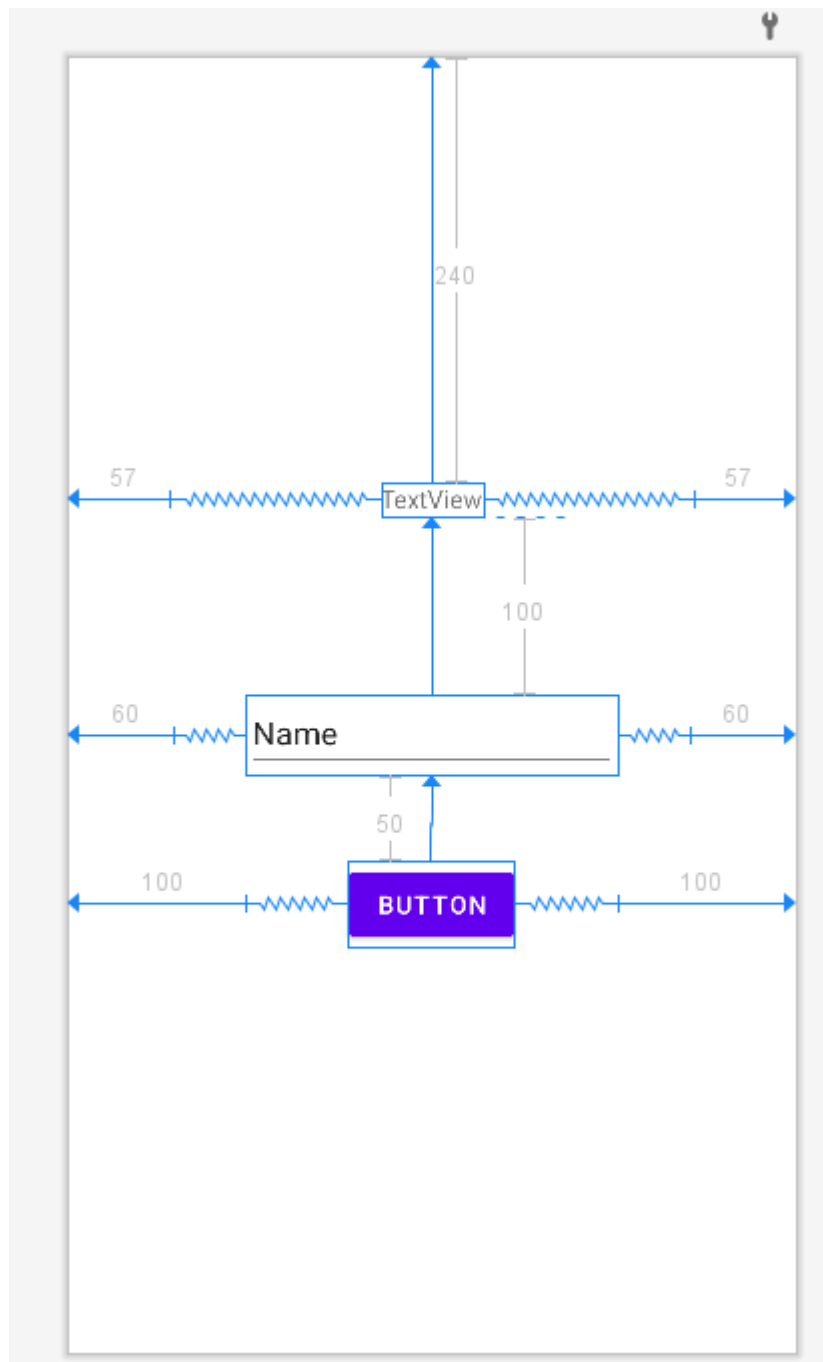
верхнее поле - 50 dp

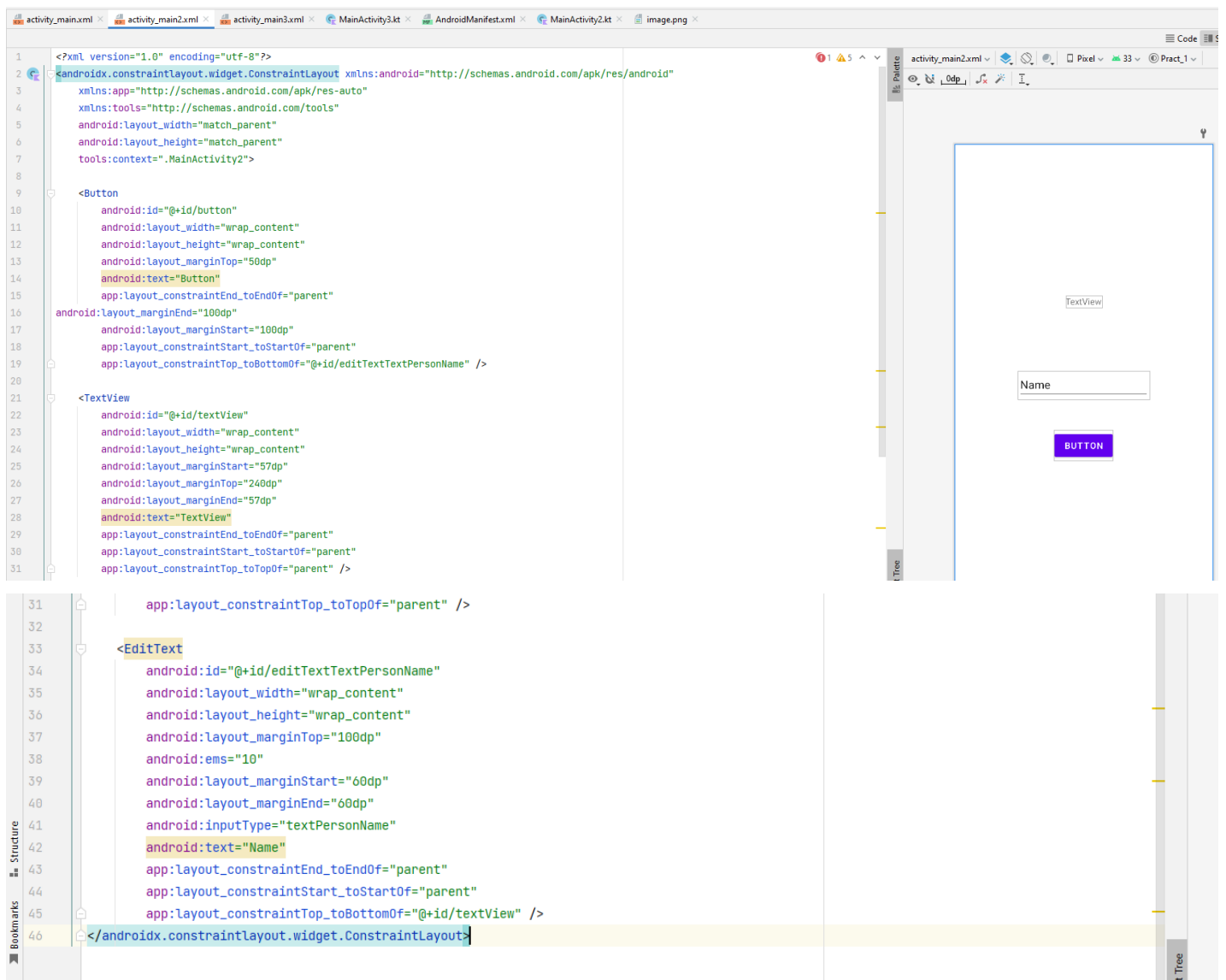
левое поле - 100 dp

правое поле - 100 dp



Настройте привязки для текстового поля `EditText`: верхняя сторона к нижней стороне объекта `TextView`, правая и левая стороны к правой и левой стороне родителя. Верхнее поле 100 dp, правое и левое по 60 dp.





Добавьте третий экран, создав еще одну Activity.

Самостоятельное задание

Самостоятельно выполните верстку 3 экрана, разместив на нем 3 изображения, 3 чекбокса, 2 текстовых контейнера и 1 кнопку. Значения настройки полей и привязок, а также итоговый вид экрана, показаны на изображениях ниже:

