

МОБИЛЬНЫЕ И WEB-ПРИЛОЖЕНИЯ

Android studio

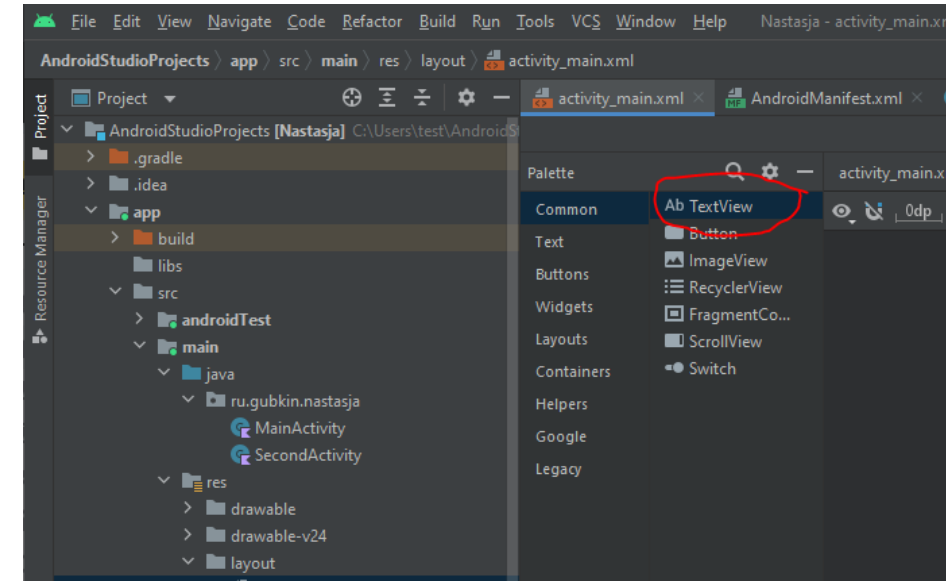
Преподаватели:

Казакова Анастасия Семеновна

Ассистент кафедры информатики

Верстка экранов

- Создаем Empty Activity.
- открываем activity_main.xml (Project/app/src/main/layout)
- Чтобы создать надпись «HelloWorld!» на экране приложения, переходим в режиме представления дизайн.
- Добавляем мышью элемент (справа от экранов) инструмент TextView. Также добавим Button с надписью «Click Me».
- Открываем текстовый редактор кода, приложение должно выглядеть так



Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:background="@color/pink"
tools:context=".MainActivity">
<TextView
    android:id="@+id/title_1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/Title_1"
    android:textColor="#2196F3"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.483"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.557" />
<Button
    android:id="@+id/button_first"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_1"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.517"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/title_1"
    app:layout_constraintVertical_bias="0.232"
    tools:ignore="MissingConstraints" />
</androidx.constraintlayout.widget.ConstraintLayout>
```



Каждая строка задает параметры элемента на экране: надпись, размеры, величину букв и т. п. Можно поиграться с параметрами, добиваясь желаемого эффекта.

Каждому элементу необходимо сделать привязку. Для этого потяните мышкой за кружок до края экрана

Activity_main.xml

Изменим фон для экрана приложения. Сейчас у нас экран белого цвета.

Возвращаемся в файл разметки **activity_main.xml**. Справа найдите вкладку **Attributes**, в которой отображаются свойства для выбранного элемента. А слева есть вкладка **Component Tree**, который отображает структуру компонентов на экране.

Необходимо выделить какой-нибудь компонент, чтобы на вкладке свойств увидеть все доступные свойства компонента.

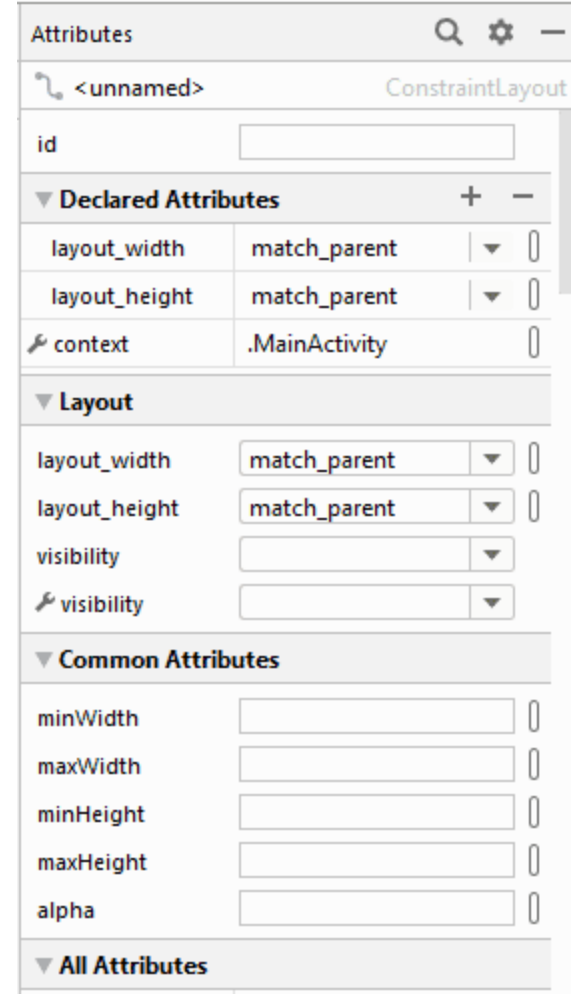
Сейчас у нас есть окно активности, графическая кнопка **ImageButton** и текстовая метка **TextView** с надписью **Hello World!**.

Так как мы собираемся работать с фоном экрана приложения, то щёлкните на **ConstraintLayout**. В панели свойств отобразятся самые употребительные свойства выбранного компонента. К ним относятся идентификатор, ширина и высота.

Находим свойство **background**. Щёлкните рядом с этим словом во второй колонке, где нужно прописывать значения.

Появится текстовое поле, в которое можно ввести значение вручную. Рядом есть маленькая кнопка, которая запустит диалоговое окно для создания ресурса.

Переходим на вкладку **Color** и выбираем цвет. Вы можете выбрать цвет, определённый в приложении (секция **app**) или в системе (секция **android**).



Если переключиться в текстовый режим, то увидим, что у элемента **ConstraintLayout** добавилась строка: `android:background="@color/colorAccent"`. Мы связали фон экран с именем цветового ресурса. Всегда используйте ресурсы.

Activity_main.xml

Добавим image на кнопку.

Находим подходящее изображение и копируем его в папку **res/drawable**.

Простое перетаскивание из проводника в папку студии не сработает. Поэтому лучше скопировать картинку в буфер, затем щёлкнуть правой кнопкой мыши на папке **drawable** в студии, выбрать команду "Вставить". Сначала появится первое диалоговое окно для выбора папки, выбираем папку **drawable**.

Выделяем элемент **ImageButton** на форме и в панели свойств выбираем свойство **srcCompat**, щёлкаем на кнопку и выбираем в диалоговом окне ресурс в категории **Drawable** - вы там должны увидеть ресурс *pinkhellokitty* (имя добавленного ранее файла).

MainActivity.kt

1. Далее откроем класс MainActivity и добавим для обработки нажатие и после-нажатие открывается следующий экран.
2. Создадим 2ю активити (File/new/activity/empty Activity). Даем название SecondActivity. Создается файл .kt и файл разметки .xml
3. Чтобы осуществить переход на другую страницу, нам нужно обеспечить доступ к нашей вью, с которой будет переход из MainActivity. Для этого:
 - в файле разметки activity_main нашей вью нужно добавить id (@ +id/имя)
 - Методом findViewById найти нашу вью
 - Обработать клик методом setOnClickListener, туда передадим функцию, которая каждый раз вызывается. Внутри этой функции вызываем метод openSecondActivity

```
package ru.gubkin.nastasja

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.TextView

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val text: TextView = findViewById<TextView>(R.id.title_1)

        text.setOnClickListener {
            openSecondActivity()
        }
        private fun openSecondActivity() {
            val intent = Intent(this, SecondActivity::class.java)
            startActivity(intent)
        }
    }
}
```

Верстка Activity_second

- В файле Activity_second.xml во вкладке дизайн добавляем нужные нам элементы : TextView, Button, Image
- Делаем настройки элементов
- Делаем привязку. К краям или друг к другу

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout 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"
    tools:context=".SecondActivity">
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/second"
        android:textColor="#3F51B5"
        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.091" />
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="130dp"
        android:layout_height="128dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.088"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView2"
        app:layout_constraintVertical_bias="0.034"
        app:srcCompat="@drawable/cat1" />
    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="129dp"
        android:layout_height="120dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.088"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.263"
        app:srcCompat="@drawable/cat2" />
    <ImageView
        android:id="@+id/imageView3"
        android:layout_width="133dp"
        android:layout_height="110dp"
        android:layout_marginTop="16dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.089"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView"
        app:layout_constraintVertical_bias="0.0"
        app:srcCompat="@drawable/cat3" />
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="40dp"
        android:text="@string/Text_second_1"
        app:layout_constraintStart_toEndOf="@+id/imageView2"
        app:layout_constraintTop_toTopOf="@+id/imageView2" />
    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="39dp"
        android:text="@string/Text_second_2"
        app:layout_constraintStart_toEndOf="@+id/imageView"
        app:layout_constraintTop_toTopOf="@+id/imageView" />
    <TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="36dp"
        android:layout_marginTop="4dp"
        android:text="@string/Text_second_3"
        app:layout_constraintStart_toEndOf="@+id/imageView3"
        app:layout_constraintTop_toTopOf="@+id/imageView3" />
</androidx.constraintlayout.widget.ConstraintLayout>
```