

Activity_main.xml

```
<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:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/img"
tools:context=".MainActivity">
```

```
<TextView
    android:id="@+id/tv1"
    android:layout_width="348dp"
    android:layout_height="55dp"
    android:layout_gravity="center"
    android:background="@color/green"
    android:gravity="center"
    android:text="@string/str1"
    android:textAllCaps="false"
    android:textColor="#FF000000"
    android:textSize="35sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.492"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.344" />
```

```
<EditText
    android:id="@+id/editText"
    android:layout_width="284dp"
    android:layout_height="87dp"
    android:layout_margin="48dp"
    android:layout_marginTop="32dp"
    android:hint="@string/str2"
    android:textColor="#FF000000"
    app:layout_constraintEnd_toEndOf="parent"
```

```
app:layout_constraintHorizontal_bias="0.493"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/tv1" />
```

```
<Button
    android:id="@+id/my_button"
    style="@style/btn_style"
    android:text="Начать"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editText"
    app:layout_constraintVertical_bias="0.309" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

Strings.xml

```
<resources>
    <string name="app_name">My Application1</string>
    <string name="str1">Добро пожаловать!</string>
    <string name="str2">Введите свое имя</string>

    <style name="btn_style">
        <item name="android:layout_width">wrap_content</item>
        <item name="android:layout_height">wrap_content</item>
        <item name="android:textColor">@color/black</item>
        <item name="android:textSize">25sp</item>
        <item name="android:textAllCaps">false</item>
        <item name="android:gravity">center</item>
        <item name="android:layout_gravity">center</item>
        <item name="android:background">#AFA3C3</item>
        <item name="android:textStyle">bold | italic</item>
        <item name="android:padding">20dp</item>
        <item name="android:typeface">serif</item>
        <item name="android:shadowColor">@color/white</item>
        <item name="android:shadowDx">10</item>
        <item name="android:shadowDy">10</item>
        <item name="android:shadowRadius">10</item>
    </style>
</resources>
```

```
</style>
</resources>
```

colors.xml

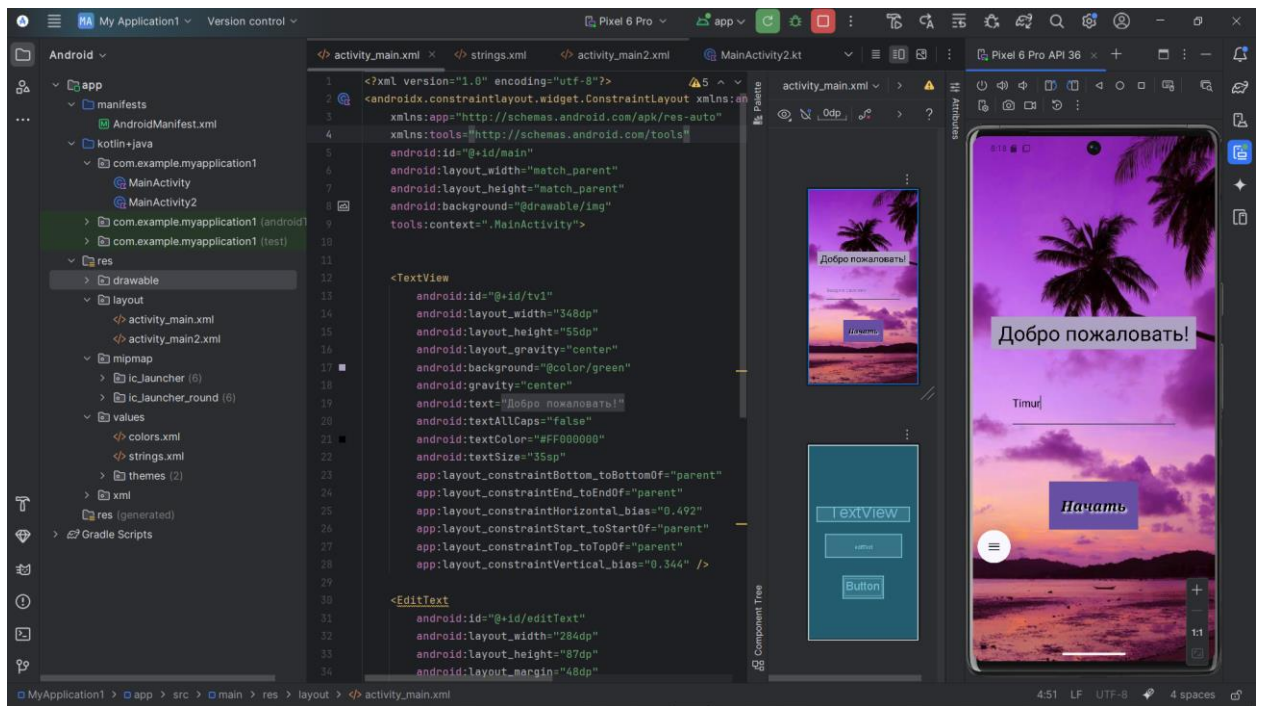
```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="green">#AFA3C3</color>
</resources>
```

Activity_main_kt

```
package com.example.myapplication1
```

```
import android.os.Bundle
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
```

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
            insets
        }
    }
}
```



Activity_main2.xml

```
<?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:orientation="vertical"
```

```
    android:gravity="center"
```

```
    android:padding="16dp"
```

```
    tools:context=".MainActivity2">
```

```
<TextView
```

```
    android:id="@+id/greetingText"
```

```
    android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:textSize="24sp"
        android:textColor="@color/color_blue"
        android:textStyle="bold"/>
```

```
<TextView
    android:id="@+id/randomNumberText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:textSize="18sp"/>
</LinearLayout>
```

Activity_main2_kt

```
package com.example.myapplication

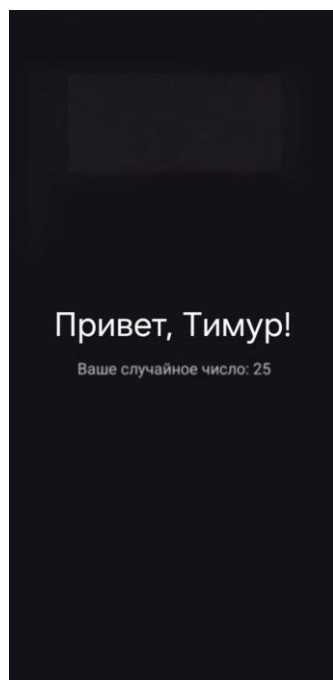
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import android.widget.TextView

class MainActivity2 : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main2) // Исправлено имя макета

        val greetingText = findViewById<TextView>(R.id.greetingText)
        val randomNumberText = findViewById<TextView>(R.id.randomNumberText)
```

```
        val userName = intent.getStringExtra("USER_NAME") ?: "User" // Исправлено на  
        getStringExtra  
  
        val randomNumber = intent.getIntExtra("RANDOM_NUMBER", 0)  
  
        greetingText.text = getString(R.string.greeting_message, userName)  
        randomNumberText.text = "Случайное число: $randomNumber" // Исправлено на text =  
    }  
}
```



Задание 2

Разница между findViewById и ViewBinding

1. Безопасность типов: ViewBinding создает привязку для каждого элемента интерфейса, что снижает вероятность ошибок, связанных с неверными идентификаторами.
2. Упрощение кода: С помощью ViewBinding вы можете обращаться к элементам интерфейса напрямую через созданные свойства, что делает код более чистым и понятным.
3. Отсутствие необходимости в вызове findViewById: Это уменьшает количество строк кода и повышает его читаемость.
4. Избежание ошибок времени выполнения: Если вы используете ViewBinding, то ошибки, связанные с отсутствующими элементами интерфейса, будут обнаружены на этапе компиляции, а не во время выполнения.