

```
tools:ignore="missingConstraints" />

<EditText
    android:id="@+id/editText"
    android:layout_width="160dp"
    android:layout_height="162dp"
    android:layout_margin="40dp"

    android:background="@drawable/fon"
    android:gravity="center"
    android:hint="Введите свое имя"
    android:padding="20dp"
    android:textStyle="bold|italic"

    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.777" />

<Button
    android:id="@+id/btn_1"
    android:layout_width="275dp"
    android:layout_height="53dp"
    android:layout_marginEnd="76dp"
    android:text="Начать"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@id/editText"
    app:layout_constraintVertical_bias="0.354" />

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

```
1 package com.example.myapplication
2
3 > import ...
4
11
12 class Activity2 : AppCompatActivity() {
13     override fun onCreate(savedInstanceState: Bundle?) {
14         super.onCreate(savedInstanceState)
15         enableEdgeToEdge()
16         setContentView(R.layout.activity2)
17         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
18             val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
19             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
20             insets
21         }
22         val text: TextView = findViewById(R.id.textView)
23         val message = intent.getStringExtra(name: "text")
24         text.text = message
25     }
26
27
28
29
```

```
1 package com.example.myapplication
2
3 import android.content.Intent
4 import android.os.Bundle
5 import android.widget.Button
6 import androidx.activity.enableEdgeToEdge
7 import androidx.appcompat.app.AppCompatActivity
8 import androidx.core.view.ViewCompat
9 import androidx.core.view.WindowInsetsCompat
10
11 class MainActivity : AppCompatActivity() {
12     override fun onCreate(savedInstanceState: Bundle?) {
13         super.onCreate(savedInstanceState)
14         enableEdgeToEdge()
15         setContentView(R.layout.activity_main)
16         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
17             val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
18             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
19             insets
20         }
21         val btn: Button = findViewById(R.id.btn_1)
22         btn.setOnClickListener {
23             val intent = Intent(packageContext: this, Activity2::class.java)
24             intent.putExtra(name: "text", value: "izmenite text")
25             startActivity(intent)
26         }
27     }
28
29
30
```

Ключевые отличия:

Эффективность: ViewBinding работает быстрее, так как привязка происходит во время компиляции, а не во время выполнения.

Безопасность: ViewBinding помогает избежать ошибок типа ClassCastException, так как свойства

привязки уже типизированы.

Чистота кода: `ViewBinding` делает ваш код более читаемым и организованным, так как вы не загромождаете его вызовами `findViewById()`.

Зачем нужен `ViewBinding`?

1. Улучшенная читаемость и удобство: Вместо того, чтобы постоянно писать `findViewById()`, вы получаете доступ к `View` из XML-файла макета через свойства.
2. Предотвращение ошибок: `ViewBinding` помогает избежать ошибок при работе с `View`, так как привязка происходит во время компиляции.
3. Эффективность: `ViewBinding` улучшает производительность, так как привязка происходит на этапе компиляции.