

Приложение  
к РППМ ПМ.01. РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ  
КОМПЬЮТЕРНЫХ СИСТЕМ  
*09.02.07 Информационные системы и программирование*

Министерство образования Московской области

ЛАБОРАТОРНАЯ РАБОТА  
«ИЗМЕНЕНИЕ ЭЛЕМЕНТОВ ДИЗАЙНА»

*Выполнено преподавателем Шкуратовой Софьей Сергеевной*

*Серпухов, 2025*

## Тема: Изменение элементов дизайна в Android Studio (Kotlin)

### Цель работы:

- познакомиться с механизмами изменения свойств элементов пользовательского интерфейса (View) в Android-приложении средствами Kotlin-кода и XML-разметки;
- научиться программно изменять внешний вид элементов: цвет, текст, размеры, отступы, параметры выравнивания и реагировать на действия пользователя.

## 1. Теоретическая часть

### 1.1. Структура интерфейса Android-приложения

В Android интерфейс создаётся с помощью:

1. XML-разметки: файл `activity_main.xml` описывает расположение и свойства элементов.

2. Программного управления: код в `MainActivity.kt` позволяет динамически менять внешний вид элементов во время выполнения приложения.

Обычно внешний вид задаётся в XML, а изменения — в Kotlin.

### 1.2. Основные элементы интерфейса (View)

Наиболее часто используемые View:

- `TextView` — текст
- `EditText` — поле ввода
- `Button` — кнопка
- `ImageView` — изображение
- `LinearLayout` / `ConstraintLayout` — контейнеры

Каждый элемент имеет набор свойств:

- цвет текста
- цвет фона
- отступы (`margin/padding`)
- размеры (`width / height`)
- выравнивание
- текстовое содержимое
- скругление углов и т. д.

### 1.3. Изменение элементов дизайна через XML

Пример `TextView` в XML:

```
<TextView
    android:id="@+id/myText"
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Hello!"
android:textSize="24sp"
android:textColor="@android:color/holo_blue_dark"
android:layout_margin="16dp"/>
```

Рис. 1 – xml-разметка

В XML можно задать:

- text — текст элемента
- textColor — цвет текста
- background — цвет или drawable
- layout\_margin — внешние отступы
- padding — внутренние отступы
- gravity — выравнивание внутри элемента

#### 1.4. Изменение элементов через Kotlin

Чтобы изменить элементы в Kotlin, нужно найти элемент по ID:

```
val text: TextView = findViewById(R.id.myText)
```

Рис. 2 – изменение элемента Kotlin

Примеры изменений:

##### 1) Изменение текста:

```
text.text = "Новый текст"
```

Рис. 3 – Изменение текста через код

##### 2) Изменение цвета текста:

```
text.setTextColor(Color.RED)
```

Рис. 4 – Изменение цвета текста через код

##### 3) Изменение цвета фона:

```
text.setBackgroundColor(Color.YELLOW)
```

Рис. 5 – Изменение цвета фона через код

##### 4) Изменение размера текста:

```
text.textSize = 30f
```

Рис. 6 – Изменение размера текста через код

##### 5) Изменение отступов:

```
val params = text.layoutParams as ViewGroup.MarginLayoutParams
```

```
params.setMargins(50, 50, 50, 50)  
text.layoutParams = params
```

**Рис. 7 – Изменение отступов текста через код**

**6) Реакция на нажатие:**

```
button.setOnClickListener {  
    text.text = "Вы нажали кнопку!"  
}
```

## 2. Практическая часть (выполнение студентом)

Студент должен создать приложение с 1 экраном, содержащим несколько элементов интерфейса и изменяющих своё состояние по нажатию кнопок.

### **Перечень заданий (для выполнения в Android Studio)**

Создать проект с активностью MainActivity.

Добавить на экран следующие элементы:

- TextView
- Button
- ImageView
- EditText
- Любой Layout (LinearLayout или ConstraintLayout)

Студент должен выполнить следующие задания:

#### **Задание 1. Изменение текста**

Сделать кнопку, по нажатию которой TextView меняет текст на любой другой.

#### **Задание 2. Изменение цвета**

Сделать две кнопки:

- кнопка 1 — меняет цвет текста TextView
- кнопка 2 — меняет цвет фона TextView

#### **Задание 3. Изменение размеров**

По нажатию кнопки увеличить размер текста TextView на 2sp.

#### **Задание 4. Изменение изображения**

При нажатии кнопки заменить изображение в ImageView на другое (заранее добавить 2 изображения в drawable).

#### **Задание 5. Изменение отступов**

По нажатию кнопки увеличить внешний отступ TextView.

#### **Задание 6. Изменение Layout параметров**

Изменить расположение TextView в Layout

(например, переместить его вниз или вправо, изменив Margin).

### **Задание 7. Работа с EditText**

Сделать кнопку «Ввести», которая переносит текст из EditText в TextView.

### **Задание 8. Свой вариант**

Студент добавляет любое ещё одно действие по изменению дизайна, например:

- сккрытие / показ элемента
- поворот изображения
- изменение прозрачности
- смена шрифта
- закруглённые углы
- анимация элемента

### 3. Что должно быть в отчёте

Отчёт должен содержать:

#### 1. Титульный лист

- название работы
- ФИО
- группа
- преподаватель
- дата

#### 2. Цель работы

#### 3. Теоретическая часть

Краткое описание:

- какие элементы интерфейса были использованы
- как в Android изменяются свойства элементов
- примеры кода `setOnClickListener`, изменения цвета, размера и т. д.

#### 4. Практическая часть

Скриншоты интерфейса и фрагменты кода:

1. XML разметка
2. Код Kotlin, который изменяет элементы
3. Скриншоты до и после нажатия кнопок

#### 5. Выводы

Что студент узнал:

- научился работать с View
- менять свойства элементов
- реагировать на действия пользователя



#### 4. Контрольные вопросы

1. Что такое View и Layout в Android?
2. Какие элементы интерфейса существуют в Android?
3. Для чего используется файл `activity_main.xml`?
4. Как получить доступ к элементу интерфейса в Kotlin-коде?
5. Какие свойства `TextView` можно менять программно?
6. Чем отличается `margin` от `padding`?
7. Как задать обработчик нажатия кнопки в Kotlin?
8. Как изменить изображение в `ImageView`?
9. Что такое ресурсы (resources) в Android?
10. Почему важно отделять XML-разметку от кода активности?