

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
«ОБРАБОТКА СОБЫТИЙ: ЦВЕТОВАЯ ИНДИКАЦИЯ»

Выполнено преподавателем Шкуратовой Софьей Сергеевной

Серпухов, 2025

Тема: Обработка событий: цветовая индикация в Android Studio (Kotlin)

Цель работы:

- изучить способы изменения цвета элементов пользовательского интерфейса в Android-приложении при обработке различных событий: нажатие, долгий клик, получение/потеря фокуса, ввод текста, смена состояния кнопок и т.п.
- Освоить применение цветов для визуальной индикации состояний элементов.

1. Теоретическая часть

Цветовая индикация — это изменение цветов элементов UI для визуальной обратной связи пользователю.

В Android изменение цвета можно выполнять:

- через XML (статически)
- через Kotlin-код (динамически)
- через ColorStateList (в зависимости от состояния)
- через события (клик, фокус, ввод текста и т. д.)

Рассмотрим разные варианты.

1.1. Изменение цвета через XML

Пример изменения цвета текста:

```
android:textColor="@android:color/holo_red_dark"
```

Рис. 1

Изменение цвета фона:

```
android:background="@android:color/holo_blue_light"
```

Рис. 2

Использование своих цветов:

В res/values/colors.xml

```
<color name="myRed">#FF4444</color>
```

Рис. 3

В XML:

```
android:textColor="@color/myRed"
```

Рис. 4

1.2. Изменение цвета через Kotlin

Подключение цвета:

```
val red = ContextCompat.getColor(this, R.color.myRed)
```

Рис. 5

Изменение цвета текста:

```
textView.setTextColor(red)
```

Рис. 6

Изменение цвета фона:

```
button.setBackgroundColor(Color.YELLOW)
```

Рис. 7

Изменение цвета при нажатии:

```
button.setOnClickListener {  
    button.setBackgroundColor(Color.GREEN)  
}
```

Рис. 8

1.3. ColorStateList (цвет зависит от состояния)

Создаём файл res/color/button_colors.xml:

```
<?xml version="1.0" encoding="utf-8"?>  
<selector xmlns:android="http://schemas.android.com/apk/res/android">  
    <item android:color="@color/myRed" android:state_pressed="true"/>  
    <item android:color="@color/myBlue"/>  
</selector>
```

Рис. 9

Используем:

```
android:textColor="@color/button_colors"
```

Рис. 10

Теперь текст меняет цвет при нажатии.

1.4. Изменение цвета при событиях

1) Короткое нажатие (click)

```
view.setOnClickListener {  
    view.setBackgroundColor(Color.GREEN)  
}
```

Рис. 11

2) Долгое нажатие (long click)

```
view.setOnLongClickListener {  
    view.setBackgroundColor(Color.RED)  
    true  
}
```

Рис. 12

3) Получение и потеря фокуса

```
editText.setOnFocusChangeListener { _, hasFocus ->  
    if (hasFocus)  
        editText.setBackgroundColor(Color.CYAN)  
    else  
        editText.setBackgroundColor(Color.WHITE)  
}
```

Рис. 13

4) Изменение цвета при вводе текста

```
editText.addTextChangedListener {  
    button.setBackgroundColor(Color.YELLOW)  
}
```

Рис. 14

5) Изменение цвета при переключателях

```
switch1.setOnCheckedChangeListener { _, isChecked ->  
    layout.setBackgroundColor(if (isChecked) Color.GREEN else Color.GRAY)  
}
```

Рис. 15

2. Практическая часть

Студент создаёт приложение с одним экраном, содержащим:

- TextView
- EditText
- Button
- ImageView
- Switch или CheckBox
- Layout (LinearLayout / ConstraintLayout)

Перечень заданий

Все задания выполняются в одном приложении.

Задание 1. Цвет при нажатии

Сделать кнопку, которая при нажатии меняет цвет фона.

Задание 2. Цвет при долгом удержании

При долгом удержании кнопки — менять её цвет текста.

Задание 3. Цветовая индикация фокуса

Сделать EditText:

- при получении фокуса → фон жёлтый
- при потере фокуса → белый

Задание 4. Цвет при вводе текста

Если пользователь вводит текст:

- TextView меняет цвет текста на синий

Задание 5. Цвет переключателя

Сделать Switch:

- ON → изменить цвет фона Layout на зелёный
- OFF → фон серый

Задание 6. Цвет картинки

При нажатии на ImageView менять оттенок изображения (фильтр цвета):

`imageView.setColorFilter(Color.RED)`

Задание 7. Цвет по кнопке Reset

Добавить кнопку «Сброс», которая возвращает все цвета в исходное состояние.

Задание 8. Личное расширение задания

Студент добавляет:

анимацию изменения цвета,

плавное затемнение/осветление,

переключение темы (светлая/тёмная),

изменение цвета при свайпе.

3. Содержание отчёта

В отчёте обязательно должны быть:

1. Титульный лист

ФИО, группа, тема, дата, преподаватель.

2. Цель работы

3. Теоретическая часть

Краткое описание:

- что такое обработка событий
- какие события влияют на цвет
- Color, ColorStateList, backgroundTint
- примеры изменения цвета

4. Практическая часть

Включает:

4.1. XML-разметка

Отрывки разметки элементов, где указаны цвета.

4.2. Kotlin-код

Код для каждого задания:

- обработчики кликов
- реакция на фокус
- реакция на ввод текста
- изменение цвета Layout

4.3. Скриншоты

“До” и “После” изменения цвета.

5. Выводы

Что было изучено:

- способы изменения цвета
- реакции элементов на события
- принципы визуальной индикации

4. Контрольные вопросы

1. Какие способы изменения цвета есть в Android?
2. Как изменить цвет текста у TextView через Kotlin?
3. Как изменить цвет фона у Button?
4. Для чего используется ColorStateList?
5. Как обработать короткое нажатие?
6. Как обработать долгий клик?
7. Как обработать изменение фокуса?
8. Как изменить цвет Layout при переключении Switch?
9. Какое событие отвечает за ввод текста в EditText?
10. Что делает метод setColorFilter у ImageView?
11. Где хранятся пользовательские цвета?
12. Чем отличается цвет текста от tint?