



# <TeachMeSkills/>

Школа программирования  
[teachmeskills.com](http://teachmeskills.com)





курс

# Android разработчик

Занятие 15. Введение в Android. Часть 4





## Агenda занятия

DataBinding, ViewBinding

Обзор принципов Material Design

Обзор основных компонентов Material Design

ViewPager, Tabs

System bars, обработка Insets



## ► **DataBinding, ViewBinding**

Обзор принципов Material Design

Обзор основных компонентов Material Design

ViewPager, Tabs

System bars, обработка Insets



## DataBinding, ViewBinding

*DataBinding официально не поддерживается Google, поэтому смысла смотреть его нет.*

*ViewBinding – это инструмент для упрощения доступа к View в коде, который заменяет `findViewById` и работает с привязкой UI-компонентов к переменным.*

*ViewBinding генерирует класс привязки для каждого layout-файла. Например, для `activity_main.xml` будет сгенерирован класс `ActivityMainBinding`.*



## ViewBinding

Чтобы включить viewBinding -> build.gradle (Module.App) в extension android добавить следующее:

```
buildFeatures { ApplicationBuildFeatures it ->
    |     viewBinding = true
    |}
```



## ViewBinding. Activity

```
private var viewBinding: ActivityMainBinding? = null

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    enableEdgeToEdge()
    viewBinding = ActivityMainBinding.inflate(layoutInflater)
    setContentView(viewBinding?.root)
```



## ViewBinding. Fragment

```
private var viewBinding: FragmentFirstBinding? = null

override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View? {
    viewBinding = FragmentFirstBinding.inflate(inflater, container, attachToParent: false)
    return viewBinding?.root
}
```



DataBinding, ViewBinding

## ► Обзор принципов Material Design

Обзор основных компонентов Material Design

ViewPager, Tabs

System bars, обработка Insets



## Обзор принципов Material Design

В 2014 году на конференции I/O была представлена новая дизайн-система, подход, который получил название *Material Design*. Новая дизайн-система позволяет создавать консistentный пользовательский опыт на всех экранах: десктоп, смартфон, планшеты, часы, телевизоры, машины. Для *Android*-приложений *Material Design* представляет собой эволюцию визуального языка *Holo* и дизайн-гайдлайнсов. Во многих смыслах это более гибкая система, которая создавалась с учетом того, что пользоваться ей будут другие дизайнеры — *Google* был лишь первым пользователем.



## Обзор принципов Material Design

Главные принципы:

- Тактильные поверхности. В Material Design интерфейс складывается из осязаемых слоёв так называемой «цифровой бумаги». Эти слои расположены на разной высоте и отбрасывают тени друг на друга, что помогает пользователям лучше понимать анатомию интерфейса и принцип взаимодействия с ним.
- Полиграфический дизайн. Если считать слои кусками «цифровой бумаги», то в том, что касается «цифровых чернил» (всего того, что изображается на «цифровой бумаге»), используется подход из традиционного графического дизайна: например, журнального и плакатного.
- Осмысленная анимация. В реальном мире предметы не возникают из ниоткуда и не исчезают в никуда — такое бывает только в кино. Поэтому в Material Design мы всё время думаем о том, как с помощью анимации в слоях и в «цифровых чернилах» давать пользователям подсказки о работе интерфейса.
- Адаптивный дизайн. Речь идет о том, как мы применяем предыдущие три концепции на разных устройствах с разными разрешениями и размерами экранов.



## Обзор принципов Material Design

Что посмотреть:

[Хорошая статья о Material Desing](#)

[Material 3 GuideLine](#)

[Material 2 GuideLine](#)



DataBinding, ViewBinding

Обзор принципов Material Design

## ► **Обзор основных компонентов Material Design**

ViewPager, Tabs

System bars, обработка Insets



## Основные компоненты Material Design

- Buttons — даже дефолтные кнопки уже будут в стиле Material
- EditText — достаточно обернуть в TextInputLayout, чтобы получить красивые анимации
- App Bars — панели навигации (Toolbar)
- Cards — контейнеры для информации, которые могут быть использованы для отображения различного контента.
- FAB (floating action button) — плавающая кнопка действия, часто используется для добавления новых элементов или действий.



DataBinding, ViewBinding

Обзор принципов Material Design

Обзор основных компонентов Material Design

## ► **ViewPager, Tabs**

System bars, обработка Insets



## ViewPager

*ViewPager — это виджет, который позволяет отображать несколько страниц (фрагментов) в одном контейнере с возможностью пролистывания между ними. Это полезно, например, для создания слайдеров или галерей.*

Основные особенности ViewPager:

- Пролистывание между фрагментами с помощью жестов (свайпы).
- Поддержка горизонтальной прокрутки.
- Работает в связке с FragmentPagerAdapter или FragmentStatePagerAdapter, которые помогают управлять фрагментами в ViewPager.



## ViewPager

атрибут orientation — вертикальная (пролистывание по вертикали), горизонтальная

```
<androidx.viewpager2.widget.ViewPager2  
    android:id="@+id/viewPager"  
    android:orientation="horizontal"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent" />
```



## ViewPager

```
class MyPagerAdapter(fragment: Fragment) : FragmentStateAdapter(fragment) {

    // Возвращает кол-во страниц
    override fun getItemCount(): Int {
        return 2
    }

    // Возвращает фрагмент для каждой страницы
    override fun createFragment(position: Int): Fragment {
        return when (position) {
            0 -> FirstFragment()
            1 -> SecondFragment()
            else -> throw IllegalStateException("Unexpected position: $position")
        }
    }
}
```

## ViewPager

*ViewPager работает в связке с адаптером, который и манипулирует нашими страницами, отвечает за их создание и прокрутку. Важно! Для правильной работы, необходимо присвоить нашему ViewPager нужный адаптер:*

```
// Привяка адаптера к ViewPager
private fun setupViewPager() {
    withBinding {
        val adapter = MyPagerAdapter(fragment: this@ThirdFragment)
        viewPager.adapter = adapter
    }
}

private fun withBinding(block: FragmentThirdBinding.() -> Unit) {
    binding?.let {
        block.invoke(it)
    }
}
```



## ViewPager. TabLayout

*Tabs* — это элемент пользовательского интерфейса, который позволяет переключаться между различными экранами или фрагментами с помощью вкладок. Это полезно для упорядочивания информации, предоставляя пользователю четкое представление о разных категориях или разделах приложения.

Для реализации вкладок часто используется *TabLayout* из библиотеки *Material Components*. *TabLayout* может работать в связке с *ViewPager*, позволяя переключаться между вкладками и отображать соответствующие фрагменты.



## ViewPager. TabLayout

*tabMode: fixed* — все вкладки будут на экране, заполняя все доступное пространство

*scrollable* — есть возможность скроллить табы

*tabGravity: center* — табы будут по центру экрана, *start* — табы будут слева, *fill* — табы займут все свободное пространство

```
<com.google.android.material.tabs.TabLayout  
    android:id="@+id/tabLayout"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    app:tabMode="fixed"  
    app:tabGravity="fill"/>
```



## ViewPager. TabLayout

Для инициализации табов, их привязки к View, модификации, используется `TabLayoutMediator()`, который принимает `TabLayoutView`, `Viewpager` и интерфейс `tabConfigurationStrategy`, у которого есть метод `onConfigureTab(position)`, где вы можете взаимодействовать с выбранным табом:

```
// Привяка адаптера к ViewPager
private fun setupViewPager() {
    withBinding {
        val adapter = MyPagerAdapter(fragment: this@ThirdFragment)
        viewPager.adapter = adapter
        // Привяка TabLayout к ViewPager
        TabLayoutMediator(tabLayout, viewPager) { tab, position ->
            // Установка текста для каждой вкладки
            tab.text = "Tab ${position + 1}"
        }.attach()
    }
}
```



DataBinding, ViewBinding

Обзор принципов Material Design

Обзор основных компонентов Material Design

ViewPager, Tabs

## ► **System bars, обработка Insets**



## System bars, обработка Insets

*System Bars — это системные элементы интерфейса, которые не принадлежат вашему приложению, но отображаются поверх него. Основные из них:*

- Status Bar — верхняя полоска, где отображаются часы, заряд батареи и иконки уведомлений.
- Navigation Bar — нижняя панель с кнопками "Назад", "Домой" и "Последние приложения" (или жестовая навигация).
- IME (Keyboard) — экранная клавиатура, тоже влияет на отображение интерфейса.



## System bars, обработка Insets

*Insets — это отступы, которые система "выделяет" под свои элементы интерфейса, чтобы ваш контент не оказался под статус-баром, навбаром или клавиатурой.*

Например:

- Когда вы открыли клавиатуру, часть вашего экрана занята — это inset от клавиатуры.
- Когда ваш контент уходит под статус-бар — это значит, вы не учли insets.

## System bars, обработка Insets

Сейчас рекомендуемый способ — использовать **WindowInsets** и **ViewCompat.setOnApplyWindowInsetsListener**:

```
ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
    val statusBarInsets = insets.getInsets(WindowInsetsCompat.Type.statusBars())
    val navBarInsets = insets.getInsets(WindowInsetsCompat.Type.navigationBars())

    v.setPadding(
        v.paddingLeft,
        statusBarInsets.top,
        v.paddingRight,
        navBarInsets.bottom
    )

    insets
}
```



## Задачи

### **Задача 1: ViewBinding: безопасный доступ к View**

Создай экран с TextView и Button. При нажатии на кнопку текст должен измениться.  
Используй ViewBinding.



## Задачи

### **Задача 2: ViewPager2 с Tabs**

Сделай экран с TabLayout и ViewPager2, в каждом табе — отдельный фрагмент.



## Задачи

### **Задача 3: Иконка и текст в TabLayout**

Добавь в TabLayout текст и иконку в одну строку.



## Задачи

### **Задача 4\*: Скрытие BottomNavigationView при клавиатуре (IME Insets)**

Скрыть BottomNavigationView, когда появляется клавиатура.



Q&A

# Домашнее задание



## Задачи

### **Задача 1: ViewPager2 с динамическими фрагментами**

Создай ViewPager2, в который можно добавлять фрагменты во время выполнения, например, при нажатии на кнопку.



## Задачи

### **Задача 2\*: Snackbar с действием**

Показать Snackbar с сообщением и кнопкой "Отмена", которая отменяет действие.



Q&A

# Ваши вопросы



# Спасибо

<TeachMeSkills/>