

Создать экран с вкладками

В файл сборки **build.gradle** модуля **app** имплементируем библиотеку:

```
implementation 'com.android.support:design:26.1.0'
```

При необходимости измените версию библиотеки на более актуальную.

Создаем адаптер

Создаем класс **MyPagerAdapter.kt**.

Класс **MyPagerAdapter** принимает экземпляр **FragmentManager** в качестве параметра и унаследуется от класса **FragmentPagerAdapter** с передачей ему экземпляра класса **FragmentManager**.

Переопределяем необходимые функции класса **FragmentPagerAdapter**: **getCount()** и **getItem()**.

В **getCount()** мы будем возвращать количество вкладок. Здесь укажем три вкладки, но вы можете указать другое необходимое количество.

В **getItem()** будем возвращать фрагмент для отображения в текущей вкладке. Позиции во вкладках начинаются с нуля, поэтому перебираем три значения: **0**, **1**, или **else** и вызываем соответствующие фрагменты, позже мы создадим и рассмотрим их код.

Переопределим функцию **getPageTitle()**, которая устанавливает заголовки вкладок. Назовем вкладки просто “First Tab”, “Second Tab”, и “Third Tab”.

Разумеется, вы можете подставить свои заголовки.

```
package ...

import android.support.v4.app.Fragment
import android.support.v4.app.FragmentManager
import android.support.v4.app.FragmentPagerAdapter

class MyPagerAdapter(fm: FragmentManager) : FragmentPagerAdapter(fm) {

    override fun getItem(position: Int): Fragment {
        return when (position) {
            0 -> {
                FirstFragment()
            }
            1 -> SecondFragment()
        }
    }
}
```

```

        else -> {
            return ThirdFragment()
        }
    }
}

override fun getCount(): Int {
    return 3
}

override fun getPageTitle(position: Int): CharSequence {
    return when (position) {
        0 -> "First Tab"
        1 -> "Second Tab"
        else -> {
            return "Third Tab"
        }
    }
}
}
}

```

Создаем фрагменты

Теперь нужно создать три фрагмента, которые будут отображаться на каждой вкладке. Если вы выбрали другое количество вкладок в вашем проекте, вам понадобится соответствующее количество фрагментов.

Создаем фрагменты с именами “FirstFragment”, “SecondFragment” и “ThirdFragment”, как мы вызывали их в классе **MyPagerAdapter**. При создании фрагментов отмечаем флажком поле “Create layout XML”, а поля “Include fragment factory methods” и “Include interface callbacks” оставляем пустыми.

В созданном автоматически для каждого фрагмента макете разметки добавляем для наглядности текст “First tab”, “Second tab” или “Third Tab” путем добавления компонента **TextView** с текстом. Это позволит при тестировании приложения определить правильность отображения фрагментов.

Вот пример кода одного из макетов **fragment_first.xml**, для остальных код аналогичен:

```

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="First tab"
    />
</FrameLayout>

```

```
        android:textSize="32sp"/>

    </FrameLayout>
```

Теперь код фрагмента **FirstFragment.kt**, для остальных фрагментов код аналогичен.

```
package ...

import android.os.Bundle
import android.support.v4.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup

class FirstFragment : Fragment() {

    override fun onCreateView(inflater: LayoutInflater?, container: ViewGroup?,
                              savedInstanceState: Bundle?): View? {
        // Inflate the layout for this fragment
        return inflater!!.inflate(R.layout.fragment_first, container, false)
    }

}
```

Главный экран

В макете главного экрана **activity_main.xml** удалим **TextView**, добавленное по умолчанию:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.design.widget.TabLayout
        android:id="@+id/tabs"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:tabMode="fixed" />

    <android.support.v4.view.ViewPager
        android:id="@+id/viewpager"
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        app:layout_constraintTop_toBottomOf="@+id/tabs"/>

</android.support.constraint.ConstraintLayout>
```

Здесь мы добавим компонент **TabLayout** с идентификатором **tabs**. Для атрибута **app:tabMode** установим параметр **fixed**. Поскольку у нас только три вкладки, нам не нужна прокрутка компонента **TabLayout**.

Также добавим компонент **ViewPager** с идентификатором **viewpager** и разместим его под **TabLayout** посредством параметра **layout_constraintTop_toBottomOf**:

```
app:layout_constraintTop_toBottomOf="@+id/tabs"
```

В классе **MainActivity.kt** подключаем **MyPagerAdapter** к нашему **ViewPager** и связываем последний с **TabLayout**. Это обеспечит привязку нужных фрагментов к их вкладкам.

```
package ...

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val fragmentAdapter = MyPagerAdapter(supportFragmentManager)
        viewpager.adapter = fragmentAdapter

        tabs.setupWithViewPager(viewpager)
    }
}
```

Тестируем приложение

Теперь запустите приложение на эмуляторе или реальном устройстве. С помощью свайпа переходите между вкладками, при этом должны отображаться соответствующие фрагменты.