TabHost и TabWidget (Вкладки)

Вкладки, как у браузеров, можно реализовать с помощью элементов **TabHost** и **TabWidget**. Компонент **TabHost** позволяет группировать связанные элементы управления в серию страниц-вкладок.

Находится в разделе **Containers**.

Хотя данный компонент не помечен, как устаревший, сама компания Google практически его не используется в своих примерах (если ошибаюсь, то ткните мордочкой в пример). В качестве альтернативы попробуйте [TabLayout (Android Design Support Library)](http://developer.alexanderklimov.ru/android/layout/tablayout.php).

**TabHost** является контейнером, который может содержать элементы **TabWidget**.

**TabHost** показывает ярлычки каждой страницы в своей коллекции. Когда пользователь выбирает вкладку, этот объект посылает сообщение в родительский контейнер **TabHost** для переключения на выбранную вкладку.

Когда вы перетаскиваете **TabHost** на форму по умолчанию, то создаётся готовый шаблон, который нужно немного модифицировать под ваши нужды - можно убрать корневой элемент**LinearLayout** и сделать таким сам **TabHost**.

<?xml version="1.0" encoding="utf-8"?>

<TabHost

android:id="@+id/tabHost"

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<TabWidget

android:id="@android:id/tabs"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

</TabWidget>

<FrameLayout

android:id="@android:id/tabcontent"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<LinearLayout

android:id="@+id/linearLayout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

</LinearLayout>

<LinearLayout

android:id="@+id/linearLayout2"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

</LinearLayout>

<LinearLayout

android:id="@+id/linearLayout3"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

</LinearLayout>

</FrameLayout>

</LinearLayout>

</TabHost>

Обратите внимание, что после элемента **TabWidget** идёт контейнер **FrameLayout**, в который вложены контейнеры **LinearLayout**, которые по сути являются формой для каждой конкретной вкладки. Поэтому здесь можно расположить ваши элементы управления: кнопки, текстовые поля и т.д.

...

<LinearLayout

android:id="@+id/tab1"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" >

<Button

android:id="@+id/button1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Кнопка на первой вкладке" />

</LinearLayout>

<LinearLayout

android:id="@+id/tab2"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" >

<Button

android:id="@+id/button2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Кнопка на второй вкладке" />

</LinearLayout>

<LinearLayout

android:id="@+id/tab3"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" >

<Button

android:id="@+id/button3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Кнопка на третьей вкладке" />

</LinearLayout>

...

Если сейчас запустить проект, то ничего хорошего не увидим. Все кнопки просто налезут друг на друга. Необходимо написать дополнительный код

package ru.alexanderklimov.tabs;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.widget.TabHost;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

setTitle("TabHost");

TabHost tabHost = (TabHost) findViewById(R.id.tabHost);

tabHost.setup();

TabHost.TabSpec tabSpec = tabHost.newTabSpec("tag1");

tabSpec.setContent(R.id.linearLayout);

tabSpec.setIndicator("Кот");

tabHost.addTab(tabSpec);

tabSpec = tabHost.newTabSpec("tag2");

tabSpec.setContent(R.id.linearLayout2);

tabSpec.setIndicator("Кошка");

tabHost.addTab(tabSpec);

tabSpec = tabHost.newTabSpec("tag3");

tabSpec.setContent(R.id.linearLayout3);

tabSpec.setIndicator("Котёнок");

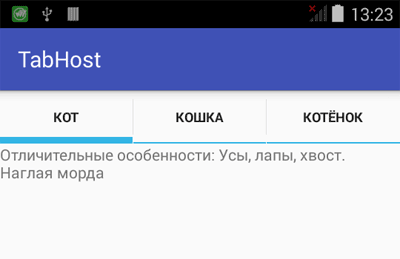
tabHost.addTab(tabSpec);

tabHost.setCurrentTab(0);

}

}

Повторно запустите проект и вы увидите вкладки, между которыми можно переключаться. Визуально выбранная вкладка имеет более жирную полоску. Причем текст на ярлыках вкладки выводится заглавными буквами, хотя в коде я писал нормальным русским языком. Магия!



Методы класса TabHost

* **setup()** — инициализирует контейнер вкладок. Необходимо вызывать перед добавлением вкладок, если **TabHost** загружается методом **findViewById()**
* **addTab()** — добавляет новую вкладку
* **setcurrentTab()** — выводит заданную вкладку на передний план

Методы класса TabWidget

Большинство методов для работы с вкладками реализованы в классе **TabWidget**. У вкладки есть индикатор позиции табуляции, информационное наполнение и идентификатор для работы в программном коде. Их необходимо определить созданием экземпляра вложенного класса **TabSpec** (см. код выше).

* **setContent()** - задает ресурс для вкладки
* **setIndicator()** - выводит текст на ярлычке вкладки

Дополнительное чтение

[Изменяем размер текста в заголовке TabWidget](http://habrahabr.ru/post/150282/) - обратите внимание на комментарии, где даются дельные советы про использование стилей.

[How to change the font size of tabhost in android](http://stackoverflow.com/questions/5788971/how-to-change-the-font-size-of-tabhost-in-android) - на эту же тему с размерами шрифта через стили