**Перелистывание страниц и ViewPager**

**ViewPager и разделение приложения на страницы**

Нередко можно встретить приложения, которые реализуют систему перелистывания, а само приложение предстает в виде набора страниц, которые можно пролистывать влево и вправо. В приложении Android для создания подобного эффекта можно использовать элемент **ViewPager**. Для создания эффекта страниц он использует фрагменты.

Итак, создадим новый проект. Добавим в папку *res/layout* файл разметки для фрагмента, который будет представлять страницу. Назовем его **fragment\_page.xml** и определим в нем следующий код:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | <FrameLayout xmlns:android="<http://schemas.android.com/apk/res/android>"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent">        <TextView          android:id="@+id/displayText"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:textSize="22sp"          android:layout\_gravity="center" />    </FrameLayout> |

Фрагмент будет отображать текстовое поле с номером страницы.

Теперь добавим в проект сам класс фрагмента. Назовем его **PageFragment**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40 | package com.example.eugene.viewpagerapp;    import android.os.Bundle;  import android.support.v4.app.Fragment;  import android.view.LayoutInflater;  import android.view.View;  import android.view.ViewGroup;  import android.widget.TextView;    public class PageFragment extends Fragment {        private int pageNumber;        public static PageFragment newInstance(int page) {          PageFragment fragment = new PageFragment();          Bundle args=new Bundle();          args.putInt("num", page);          fragment.setArguments(args);          return fragment;      }        public PageFragment() {      }        @Override      public void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          pageNumber = getArguments() != null ? getArguments().getInt("num") : 1;      }        @Override      public View onCreateView(LayoutInflater inflater, ViewGroup container,                               Bundle savedInstanceState) {          View result=inflater.inflate(R.layout.fragment\_page, container, false);          TextView pageHeader=(TextView)result.findViewById(R.id.displayText);          String header = String.format("Фрагмент %d", pageNumber+1);          pageHeader.setText(header);          return result;      }  } |

Переменная pageNumber указывает на номер текущей страницы. Номер страницы будет передаваться извне через фабричный методnewInstance(). Передача номера происходит путем добавления значения в аргумент "num"

Затем при создании фрагмента в методе onCreate() этот номер будет извлекаться из аргумента "num" (если аргументы определены):

|  |  |
| --- | --- |
| 1 | pageNumber = getArguments() != null ? getArguments().getInt("num") : 1; |

В методе onCreateView() полученный номер страницы будет отображаться в текстовом поле.

Сам по себе фрагмент еще не создает функциональность постраничной навигации. Для этого нам нужен один из классов **PagerAdapter**. Android SDK содержит две встроенных реализации PagerAdapter: классы **FragmentPagerAdapter** и **FragmentStatePagerAdapter**.

FragmentPagerAdapter хранит все количество используемых фрагментов в памяти и поэтому подходит, если в приложении небольшое количество страниц. А класс FragmentStatePagerAdapter хранит в памяти только текущую, предыдущую и следующую страницу (при их наличии). За счет этого он экономнее расходует память, но в то же время работает чуть медленнее. В данном случае воспользуемся классом FragmentPagerAdapter.

Однако оба класса являются абстрактными, поэтому напрямую мы их использовать не можем, и нам нужно создать класс-наследник. Для этого добавим в проект новый класс, который назовем **MyAdapter**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | package com.example.eugene.viewpagerapp;    import android.support.v4.app.Fragment;  import android.support.v4.app.FragmentManager;  import android.support.v4.app.FragmentPagerAdapter;    public class MyAdapter extends FragmentPagerAdapter {      public MyAdapter(FragmentManager mgr) {          super(mgr);      }      @Override      public int getCount() {          return(10);      }      @Override      public Fragment getItem(int position) {          return(PageFragment.newInstance(position));      }  } |

Класс FragmentPagerAdapter определяет два метода:

* getCount(): возвращает количество страниц, которые будут в ViewPager (в нашем случае 10)
* getItem(int position): по номеру страницы, передаваемому в качестве параметра position, возвращает объект фрагмента

В завершении установим в файле **activity\_main.xml** элемент ViewPager:

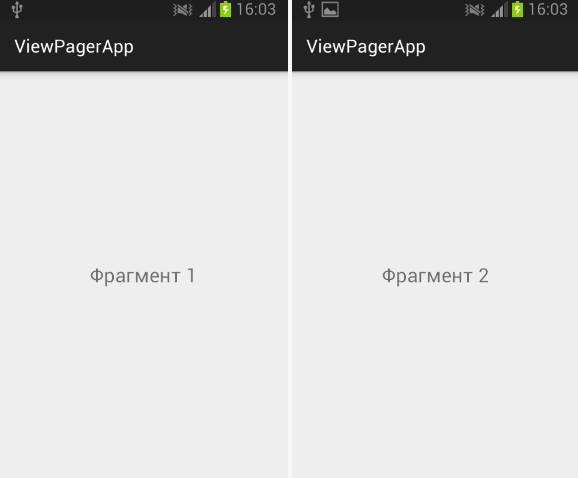
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <android.support.v4.view.ViewPager      xmlns:android="<http://schemas.android.com/apk/res/android>"      android:id="@+id/pager"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent">    </android.support.v4.view.ViewPager> |

И также изменим код **MainActivity**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | package com.example.eugene.viewpagerapp;    import android.support.v7.app.ActionBarActivity;  import android.os.Bundle;  import android.view.Menu;  import android.view.MenuItem;    import android.support.v4.view.ViewPager;    public class MainActivity extends ActionBarActivity {        @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_main);            ViewPager pager=(ViewPager)findViewById(R.id.pager);          pager.setAdapter(new MyAdapter(getSupportFragmentManager()));      }  } |

Чтобы перелистывание заработало, установим для ViewPager адаптер MyAdapter.

И запустив проект, мы сможем с помощью перелистывания перемещаться по страницам:



**Заголовки страниц, PagerTitleStrip и PagerTabStrip**

**Последнее обновление: 22.03.2017**

В прошлой теме мы рассмотели, как создать функциональность перелистывания страниц. Теперь пойдем дальше и добавим к страницам заголовки, посредством которых мы можем дополнительно перемещаться по станицам.

Для добавления заголовков мы можем использовать элементы **PagerTitleStrip** и **PagerTabStrip**. Главная разница между ними заключается в том, что заголовки в PagerTabStrip фактически представляют собой интерактивные вкладки, нажав на которые, мы можем перейти к определенной странице. А на элементе PagerTitleStrip все заголовки просто являются статическим текстом.

Используем PagerTabStrip. Для этого возьмем проект из прошлой темы.

Вначале изменим файл **activity\_main.xml**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | <android.support.v4.view.ViewPager      xmlns:android="<http://schemas.android.com/apk/res/android>"      android:id="@+id/pager"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent">        <android.support.v4.view.PagerTabStrip          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:layout\_gravity="top"/>    </android.support.v4.view.ViewPager> |

Изменим код фрагмента **PageFragment**, чтобы он мог устанавливать заголовок вкладок:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | package com.example.eugene.viewpagerapp;    import android.content.Context;  import android.os.Bundle;  import android.support.v4.app.Fragment;  import android.view.LayoutInflater;  import android.view.View;  import android.view.ViewGroup;  import android.widget.TextView;    public class PageFragment extends Fragment {        private int pageNumber;        public static PageFragment newInstance(int page) {          PageFragment fragment = new PageFragment();          Bundle args=new Bundle();          args.putInt("num", page);          fragment.setArguments(args);          return fragment;      }        public PageFragment() {      }        @Override      public void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          pageNumber = getArguments() != null ? getArguments().getInt("num") : 1;      }      static String getTitle(Context context, int position) {          return "Страница № " + String.valueOf(position+1);      }      @Override      public View onCreateView(LayoutInflater inflater, ViewGroup container,                               Bundle savedInstanceState) {          View result=inflater.inflate(R.layout.fragment\_page, container, false);          TextView pageHeader=(TextView)result.findViewById(R.id.displayText);          pageHeader.setText("Фрагмент " + String.valueOf(pageNumber+1));          return result;      }  } |

Здесь добавлен метод getTitle(), задача которого - установка заголовка страницы.

Теперь изменим код адаптера **MyAdapter**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | package com.example.eugene.viewpagerapp;    import android.content.Context;  import android.support.v4.app.Fragment;  import android.support.v4.app.FragmentManager;  import android.support.v4.app.FragmentPagerAdapter;    public class MyAdapter extends FragmentPagerAdapter {      private Context context = null;        public MyAdapter(Context context, FragmentManager mgr) {          super(mgr);          this.context = context;      }        @Override      public int getCount() {          return (10);      }        @Override      public Fragment getItem(int position) {          return (PageFragment.newInstance(position));      }        @Override      public String getPageTitle(int position) {          return (PageFragment.getTitle(context, position));      }  } |

С помощью метода getPageTitle() мы можем возвратить заголовок для вкладки в PagerTabStrip. Фактически в нем мы обращаемся к методуgetTitle() фрагмента для установки заголовка.

И в конце немного подредактируем код **MainActivity**, изменив ее метод onCreate:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | package com.example.eugene.viewpagerapp;    import android.support.v4.view.ViewPager;  import android.support.v7.app.AppCompatActivity;  import android.os.Bundle;    public class MainActivity extends AppCompatActivity {        @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_main);            ViewPager pager=(ViewPager)findViewById(R.id.pager);          pager.setAdapter(new MyAdapter(this, getSupportFragmentManager()));      }  } |

Все остальное остается без изменений. Запустим проект на выполнение и увидим интерактивные вкладки-заголовки поверх страниц.

