

Тема 9. «Создание простого меню»¹

Android поддерживает несколько типов меню.

Первый. На телефоне есть отдельная кнопка Menu (в старых телефонах), нажатие которой вызывает меню. В новых устройствах отдельную кнопку убрали, заменив на значок меню в виде трёх точек в вертикальной ориентации.

Второй тип – контекстное меню, которое появляется при нажатии и удерживания пальца на экране в нужном месте (также можно нажать и удерживать центральную кнопку на телефоне). Контекстное меню в свою очередь может иметь подменю. Сейчас мы познакомимся с первым типом меню.

Меню уже встроено по умолчанию при создании нового проекта. Посмотрим, как оно работает.

Создайте новый проект и запустите его. Нажмите на значок меню в правой части заголовка, чтобы увидеть пункт меню Settings, рис. 1.

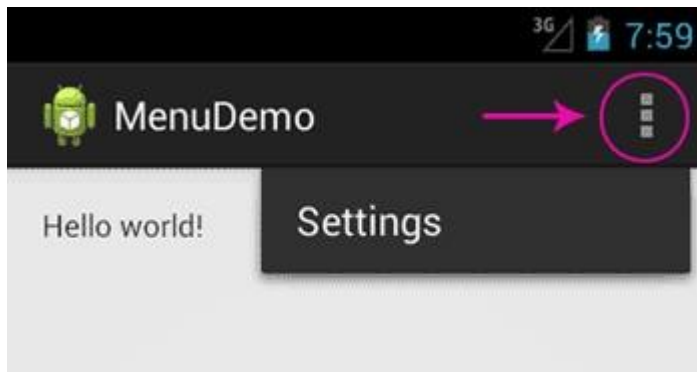


Рис. 1. Пункт меню Settings

Попробуем разобраться, как это устроено.

Само слово Settings хранится в файле `res/values/strings.xml` под именем `action_settings`. Давайте создадим ещё три новых элемента по такому же принципу:

```
<string name="action_cat_male">Кот</string>
<string name="action_cat_female">Кошка</string>
<string name="action_kitten">Котёнок</string>
```

Теперь найдите папку `res/menu`. В ней есть файл `menu_main.xml` – имя указывает, что меню относится к основной активности `MainActivity`. Если вы будете создавать приложение с несколькими экранами, то у каждой активности будет отдельное меню со своими настройками. Пока откроем файл `menu_main.xml` для изучения:

Для Android 4.0

<menu xmlns:android="http://schemas.android.com/apk/res/android" >
<item
android:id="@+id/action_settings"
android:orderInCategory="100"
android:showAsAction="never"
android:title="@string/action_settings"/>
</menu>

В Android 5.0 имеются небольшие отличия. Атрибут showAsAction теперь относится к пространству имён app.

<menu xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
tools:context=".MainActivity">
<item
android:id="@+id/action_settings"
android:title="@string/action_settings"
android:orderInCategory="100"
app:showAsAction="never" />
</menu>

Элемент item отвечает за отдельный пункт меню. Добавим ещё три пункта по такому же принципу, меняя только идентификатор и текст для меню:

<menu xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
tools:context=".MainActivity">
<item
android:id="@+id/action_settings"
android:title="@string/action_settings"
android:orderInCategory="100"
app:showAsAction="never" />
<item
android:id="@+id/action_cat1"
android:orderInCategory="100"
app:showAsAction="never"
android:title="@string/action_cat_male" />
<item
android:id="@+id/action_cat2"
android:orderInCategory="100"
app:showAsAction="never"
android:title="@string/action_cat_female" />
<item
android:id="@+id/action_cat3"
android:orderInCategory="100"
app:showAsAction="never"
android:title="@string/action_kitten" />
</menu>

Запустите проект и попробуйте снова вызвать меню. Вы увидите три новых пункта, рис. 2.

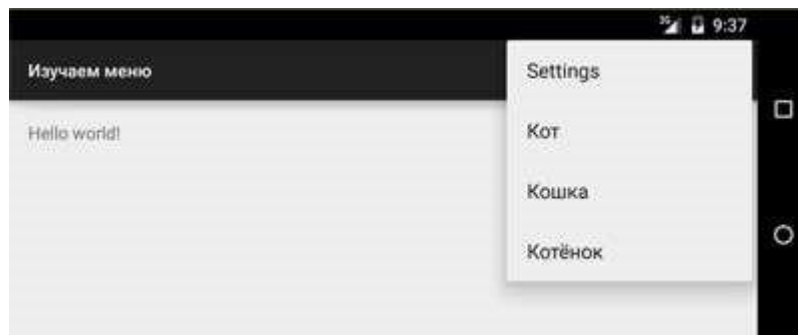


Рис. 2. Отображение пунктов

Атрибут `app:showAsAction="never"` указывает, что пункты меню никогда не должны выводиться в заголовке приложения, а всегда находиться за тремя точками. Если заменить на значение `ifRoom`, то пункт сразу выведется в заголовке. Если таких пунктов будет много, то они будут только мешаться. Как правило, в таком варианте выводят очень короткое слово или значок для частых операций, чтобы избежать лишнего щелчка на три точки.

Пока пункты меню не выполняют полезной работы. Любое нажатие на пункт просто закрывает меню без видимых последствий. Мы ещё не написали код для обработки нажатий. Откроем файл `MainActivity.java` и изучим его.

В нём вы уже видели код для метода `onCreateOptionsMenu()`. Именно данный метод отвечает за появление меню у активности. Закомментируйте вызов метода и убедитесь, что меню не будет выводиться.

```
@Override  
  
public boolean onCreateOptionsMenu(Menu menu) {  
  
    getMenuInflater().inflate(R.menu.menu_main, menu);  
  
    return true;}  

```

В методе `inflate()` вы указываете на ресурс меню (`R.menu.menu_main`) и объект класса `Menu`. Вся подготовительная работа сделана за вас и вам не нужно ничего помнить.

Мы научились создавать меню. Но пока оно бесполезно, так как пункты меню никак не реагируют на наши нажатия. Для обработки нажатий пунктов меню служит метод `onOptionsItemSelected()`. Данный метод тоже уже присутствует в проекте и вам не надо ничего помнить. А вот раньше программистам приходилось добавлять метод вручную.

```

@Override

public boolean onOptionsItemSelected(MenuItem item) {

    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.

    int id = item.getItemId();

    //noinspection SimplifiableIfStatement

    if (id == R.id.action_settings) {

        return true;}

    return super.onOptionsItemSelected(item);}

```

В методе надо указать идентификатор пункта меню и код, который должен выполняться для него. В заготовке используется оператор if.

Мы заменим его на switch для удобства, но вы можете использовать и конструкцию if else. Для вывода информации воспользуемся текстовой меткой. Добавьте на экран активности компонент TextView. Можете использовать имеющийся TextView с надписью «Hello World!», только присвойте ему идентификатор.

```

@Override

public boolean onOptionsItemSelected(MenuItem item) {

    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.

    int id = item.getItemId();

    TextView infoTextView = (TextView) findViewById(R.id.textViewInfo);

    // Операции для выбранного пункта меню

    switch (id) {

        case R.id.action_cat1:

            infoTextView.setText("Вы выбрали кота!");

            return true;

        case R.id.action_cat2:

            infoTextView.setText("Вы выбрали кошку!");

            return true;

        case R.id.action_cat3:

            infoTextView.setText("Вы выбрали котёнка!");

            return true;

        default:

            return super.onOptionsItemSelected(item);} }

```

Запустите приложение, вызовите меню и выберите любой пункт меню. В текстовом поле должно появиться сообщение, рис. 3.

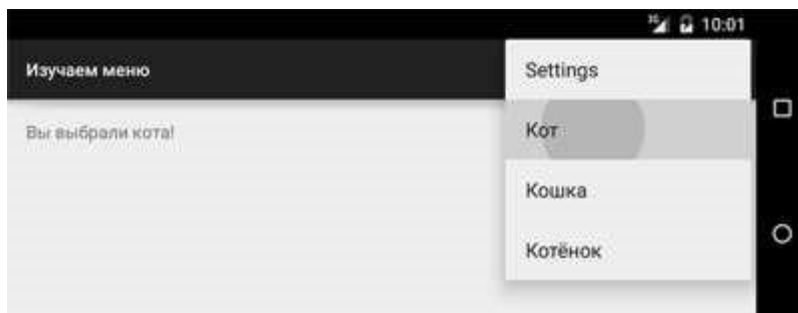


Рис. 3. Вывод сообщения

Существует альтернативный способ через XML, похожий на обработку щелчков кнопки (начиная с Android 3.0). Вы можете добавить атрибут `android:onClick` в ресурсах меню, и вам не нужно использовать вызов метода `onOptionsItemSelected()`. При помощи `android:onClick` вы можете указать нужный метод при выборе пункта меню. Добавьте данный атрибут к пункту `Settings`.

```
<item
    android:id="@+id/action_settings"
    android:title="@string/action_settings"
    android:orderInCategory="100"
    android:onClick="onSettingsMenuClick"
    app:showAsAction="never" />
```

Теперь в коде активности напишем следующее:

// у атрибута пункта меню `Settings` установлено значение `android:onClick="onSettingsMenuClick"`

```
public void onSettingsMenuClick(MenuItem item) {
    TextView infoTextView = (TextView) findViewById(R.id.textViewInfo);
    infoTextView.setText("Вы выбрали пункт Settings, лучше бы выбрали кота");}
```

Внешний вид пунктов меню можно изменить на вид с переключателями, рис. 4. Для этого нужно добавить элемент `group` с атрибутом `android:checkableBehavior="single"`:

```

<menu xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
tools:context=".MainActivity">
    <item
        android:id="@+id/action_settings"
        android:orderInCategory="100"
        android:title="@string/action_settings"
        app:showAsAction="never" />
    <group android:checkableBehavior="single">
        <item
            android:id="@+id/action_cat1"
            android:orderInCategory="100"
            android:title="@string/action_cat_male"
            app:showAsAction="never" />
        <item
            android:id="@+id/action_cat2"
            android:orderInCategory="100"
            android:title="@string/action_cat_female"
            app:showAsAction="never" />
        <item
            android:id="@+id/action_cat3"
            android:orderInCategory="100"
            android:title="@string/action_kitten"
            app:showAsAction="never" />
    </group>
</menu>

```

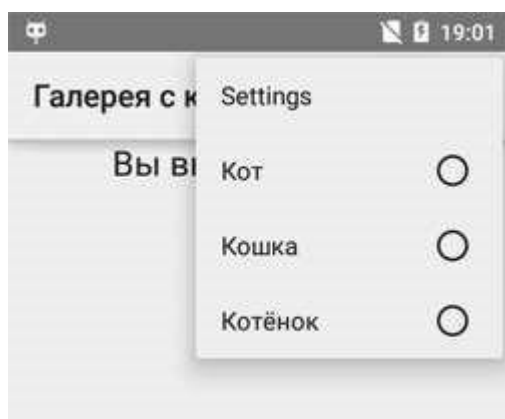



Рис. 4. Отображение пунктов меню