

[RxJava \(http://developer.alexanderklimov.ru/android/rx/\)](http://developer.alexanderklimov.ru/android/rx/)

[Советы \(http://developer.alexanderklimov.ru/android/tips-android.php\)](http://developer.alexanderklimov.ru/android/tips-android.php)

[Статьи \(http://developer.alexanderklimov.ru/android/articles-android.php\)](http://developer.alexanderklimov.ru/android/articles-android.php)

[Книги \(http://developer.alexanderklimov.ru/android/books.php\)](http://developer.alexanderklimov.ru/android/books.php)

[Java \(http://developer.alexanderklimov.ru/android/java/java.php\)](http://developer.alexanderklimov.ru/android/java/java.php)

[Kotlin \(http://developer.alexanderklimov.ru/android/kotlin/\)](http://developer.alexanderklimov.ru/android/kotlin/)

[Дизайн \(http://developer.alexanderklimov.ru/android/design/\)](http://developer.alexanderklimov.ru/android/design/)

[Отладка \(http://developer.alexanderklimov.ru/android/debug/\)](http://developer.alexanderklimov.ru/android/debug/)

[Open Source \(http://developer.alexanderklimov.ru/android/opensource.php\)](http://developer.alexanderklimov.ru/android/opensource.php)

[Полезные ресурсы \(http://developer.alexanderklimov.ru/android/links.php\)](http://developer.alexanderklimov.ru/android/links.php)

Меню

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. Сегодня мы познакомимся с первым типом меню. В данной статье будет рассматриваться работа с меню на новых устройствах под управлением Android 4.0 и выше.

В шаблоне **Empty Activity** нет меню, поэтому мы создадим его сами. Это поможет вам понять принцип работы и получить общее представление о проекте. Запоминать названия классов, методов и код для обработки выбора пунктов меню необязательно. В других шаблонах меню будет встроено и вы можете сразу использовать его.

Создайте новый проект на основе **Empty Activity** и запустите его. Никакого меню пока нет.

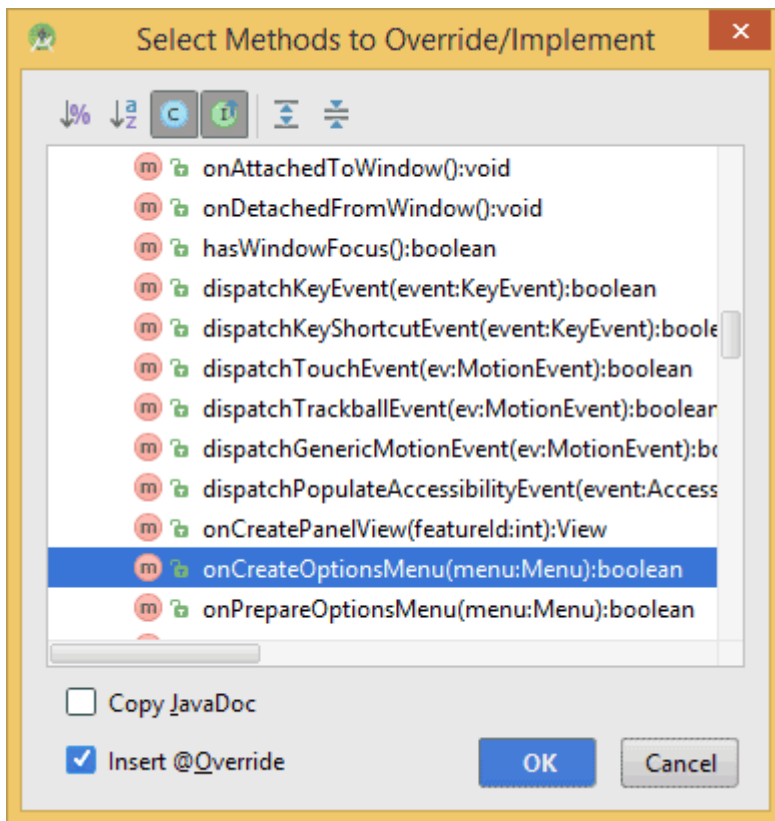
Создадим несколько строковых ресурсов в файле **res/values/strings.xml**, которые будут отвечать за пункты меню:

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

Теперь создайте новую папку **menu** в папке **res** (правый щелчок мыши на папке **res**, | **New** | **Directory**). Далее создайте в созданной папке файл **menu_main.xml** - имя указывает, что меню относится к основной активности **MainActivity** (правый щелчок мыши на папке **menu** | **New** | **Menu Resource File**). Если вы будете создавать приложение с несколькими экранами, то у каждой активности будет отдельное меню со своими настройками. Пока откроем файл **menu_main.xml** и добавим в полученный шаблон свой код:

```
<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"/>
</menu>
```

Откроем файл **MainActivity**. Сейчас в нём только один метод **onCreate()**. Добавим новый метод **onCreateOptionsMenu()**. Именно данный метод отвечает за появление меню у активности. Выберите в студии меню **Code** | **Override Methods...** и в следующем окне начинайте вводить название метода по первым буквам. Можно вводить по первым заглавным буквам, т.е. **осом** (**onCreateOptionsMenu**), чтобы быстро найти нужную строку. Нажимаем кнопку **OK** и получаем заготовку.



```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    return super.onCreateOptionsMenu(menu);
}
```

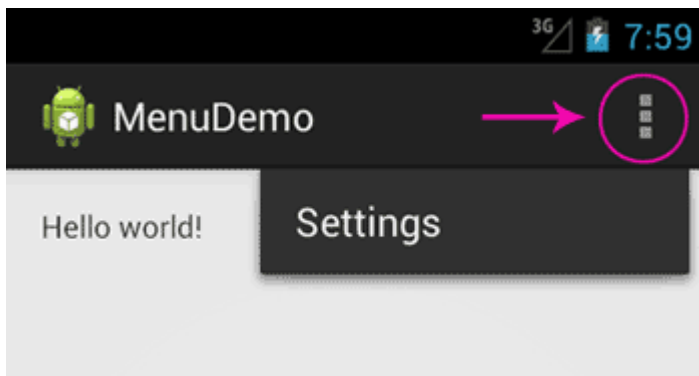
Добавляем в заготовку метод, который берёт данные из ресурсов меню и преобразует их в пункты меню на экране.

```
@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**.

По английски "inflate" переводится как надувать, т.е. по замыслу разработчиков Android, мы как бы надуваем данными объект, например, меню. Но на самом деле слово "inflate" происходит от словосочетания **in flat** - в квартиру. Существует старинная традиция запускать в квартиру первым кота, который исследует все закоулки дома и заявляет о своём согласии жить в нём. Так и мы запускаем данные из XML-файла в объект **MenuInflater**.

Запустите проект. Теперь в правой части заголовка вы увидите значок из трёх точек, выстроенных в вертикальную линию. Нажмите на значок, чтобы увидеть пункт меню **Settings**.



Как не трудно догадаться, элемент **item** отвечает за отдельный пункт меню. Добавим ещё три пункта по такому же принципу, меняя только идентификатор и текст для меню:

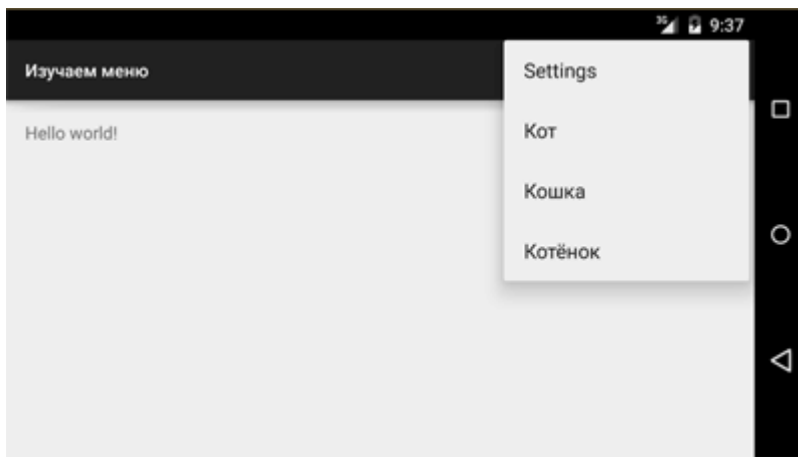
```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:tools="http://schemas.android.com/tools"
      tools:context=".MainActivity">
    <item
        android:id="@+id/action_settings"
        android:orderInCategory="100"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

    <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"/>
</menu>
```

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



Параметры **id** и **title** не нуждаются в объяснениях. Параметр **orderInCategory** позволяет задать свой порядок вывода пунктов меню. Предположим вы создали пять пунктов меню, но пока не определились с порядком их вывода на экране. Чтобы не перемещать постоянно целые блоки кода для пунктов меню в нужном порядке, можно воспользоваться данным параметром.

И, наконец, важный атрибут **app:showAsAction** определяет поведение меню в **ActionBar**. Значение **never** означает, что элемент меню не должен выводиться в заголовке, а только в всплывающем меню, т.е. находиться за тремя точками. Если вы установите значение **always**, то пункт **Settings** сразу появится в заголовке вашего приложения. Также доступны значения **ifRoom**, **withText** и **collapseActionView**. Попробуйте самостоятельно. Например, **ifRoom** выводит пункт меню, если позволяет место. Если 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.

Обратите внимание на атрибут **app:showAsAction**, который относится к пространству имён **xmlns:app="http://schemas.android.com/apk/res-auto"**. Было время, когда такого пространства имён не существовало и в проектах использовался атрибут **android:showAsAction** из стандартного пространства имён. Если студия будет ругаться на ошибку, то отредактируйте код.

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.

Выбор пунктов меню

Мы научились создавать меню. Но пока оно бесполезно, так как пункты меню никак не реагируют на наши нажатия. Для обработки нажатий пунктов меню служит другой метод **onOptionsItemSelected()**. Добавим метод по такому же принципу, как для предыдущего примера. Получим заготовку.

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    return super.onOptionsItemSelected(item);
}
```

Параметр **item** отвечает за пункт меню. Вам следует получить идентификатор меню через метод **getItemId()** и указать для него код. Так как обычно меню состоит из нескольких пунктов, то удобно использовать конструкции **if/else** или **switch**. Для вывода информации воспользуемся текстовой меткой. Добавьте на экран активности компонент **TextView**. Можете использовать имеющийся **TextView** с надписью "Hello World!", только присвойте ему идентификатор.

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"/>
```

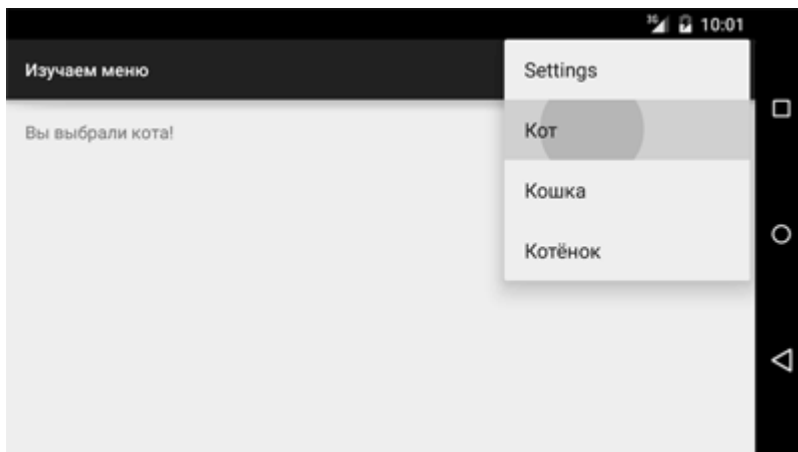
Добавим код в заготовку для выбранного пункта меню:

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // получим идентификатор выбранного пункта меню
    int id = item.getItemId();

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

    // 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.



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 напомним следующее:

```
// у атрибута пункта меню Settings установлено значение android:onClick="onSettingsMenuClick"
public void onSettingsMenuClick(MenuItem item) {
    TextView infoTextView = (TextView) findViewById(R.id.textView);
    infoTextView.setText("Вы выбрали пункт Settings, лучше бы выбрали кота");
}
```

Переключатели

Внешний вид пунктов меню можно изменить на вид с переключателями. 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" />

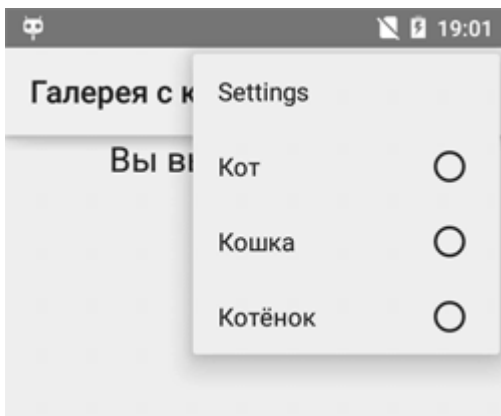
    <group android:checkableBehavior="single">

        <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>

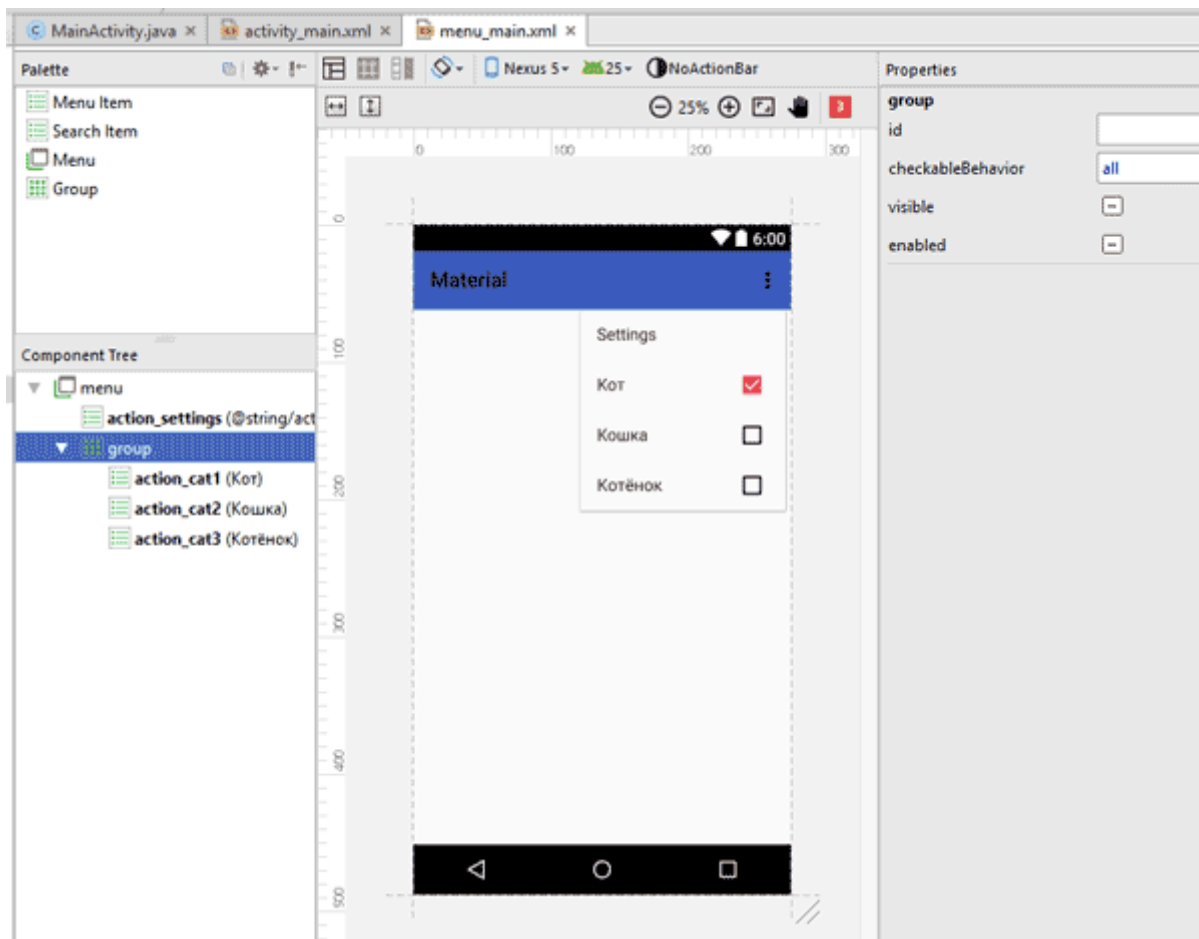
```



Большого смысла в этом режиме я не вижу. И рассматривать его не будем.

Режим Design

В Android Studio 2.2 добавили графический режим построения меню, которых похож на панель инструментов для добавления новых компонентов на экран. У меню панель состоит из четырёх элементов: **Menu Item**, **Search Item**, **Menu**, **Group**.



Принцип тот же, выбираете нужный элемент и перетаскиваете его на экран в область меню. Если вы изучили ручное создание меню, то данный способ не вызовет у вас затруднений. С его помощью можно быстро набросать структуру меню, а затем подправить вручную.

Итак, вы получили базовые навыки работы с меню, достаточных для большинства случаев. Если хотите узнать больше, то почитайте [дополнительную информацию о меню в разделе Теория \(theory/menu.php\)](http://theory.menu.php).

Дополнительное чтение

[Обсуждение статьи \(http://forum.alexanderklimov.ru/viewtopic.php?id=10\)](http://forum.alexanderklimov.ru/viewtopic.php?id=10) на форуме.

Реклама

Реклама