**[Меню](http://ru-code-android.livejournal.com/1700.html" \t "_self)**

Меню являются важной частью многих android приложений. Оно предоставляет пользователю привычный и удобный способ для выполнения каких-то действий.  
  
Есть три типа меню:

* *Меню параметров (Options Menu)* - Это меню, которое появляется, когда пользователь нажимает кнопку Menu на устройстве.
* *Контекстное меню (Context Menu)* - появляется при длительном нажатии на экран.
* *Подменю (Submenu)* - Дополнительное меню, для одного из пунктов основного меню

SDK Android предлагает удобные инструменты для создания меню.   
  
Пост оказался очень длинным, поэтому разбит на две части.  
  
Основной класс отвечающий за меню [android.view.Menu](http://developer.android.com/reference/android/view/Menu.html" \t "_self). Для элементов меню используется класс[android.view.MenuItem](http://developer.android.com/reference/android/view/MenuItem.html), а для подменю класс [android.view.Submenu](http://developer.android.com/reference/android/view/SubMenu.html" \t "_self).   
Любая деятельность (Activity) связана с объектом класса Menu. Android создает для каждой деятельности отдельное меню и передает его методу *bool onCreateOptionsMenu (Menu menu)*. Данный метод позволяет заполнить созданное меню нужными элементами. Метод возвращает true, чтобы меню было видимым и false — невидимым.  
  
Добавить элемент в меню можно с помощью метода add класса Menu:

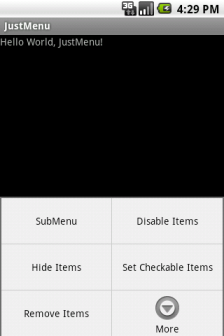
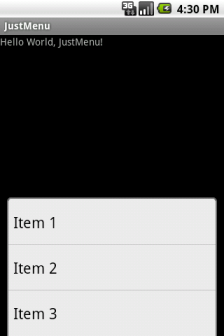
copy to clipboard[подсветка кода](http://aivolkov.ru/online-syntax-highlighter/)

1. MenuItem add (**int** groupId, **int** itemId, **int** order, charSequence Title)

где *groupId* — id группы. Элементы меню можно объединять в группы, присваивая каждой из них групповой ID. В классе Menu есть несколько методов для управления группами элементов.  
*itemId* — id конкретного элемента меню  
*order* — порядок сортировки  
*Title* — заголовок элемента меню  
  
**Создание меню**  
Создадим простое приложение, в котором постараемся рассмотреть основные принципы работы с меню. Создаем новый проект (у меня это JustMenu) и добавляем метод *onCreateOptionsMenu*

copy to clipboard[подсветка кода](http://aivolkov.ru/online-syntax-highlighter/)

1. **public** **class** JustMenu **extends** Activity {
3. Menu myMenu;
4. **boolean** isGroupVisible = **true**;
5. **boolean** isGroupEnabled = **true**;
6. **boolean** isGroupCheckable = **false**;
7. **boolean** isGroupExists = **true**;
9. @Override
10. **public** **void** onCreate(Bundle savedInstanceState) {
11. **super**.onCreate(savedInstanceState);
12. setContentView(R.layout.main);
13. }
15. @Override
16. **public** **boolean** onCreateOptionsMenu(Menu menu) {
18. myMenu = menu;
20. menu.add (Menu.FIRST, 1, 1, "SubMenu");
21. menu.add (Menu.FIRST, 2, 2, "Disable Items");
22. menu.add (Menu.FIRST, 3, 3, "Hide Items");
23. menu.add (Menu.FIRST, 4, 4, "Set Checkable Items");
24. menu.add (Menu.FIRST, 5, 5, "Remove Items");
26. menu.add (Menu.CATEGORY\_SECONDARY, 6, 6, "Item 1");
27. menu.add (Menu.CATEGORY\_SECONDARY, 7, 7, "Item 2");
28. menu.add (Menu.CATEGORY\_SECONDARY, 8, 8, "Item 3");
29. **return** **true**;
30. }
31. }

*Menu.FIRST* и *Menu.CATEGORY\_SECONDARY* - две константы в классе Menu, которые можно использовать как ID групп. О других константах можно почитать [тут](http://developer.android.com/reference/android/view/Menu.html).  
  
Мы добавили в класс переменную *myMenu* и инициализировали ее в методе *onCreateOptionsMenu*, для того чтобы можно было обращаться к меню из других мест программы, и четыре булевых переменных (их смысл будет понятен ниже). В группе *Menu.FIRST* находится пять элементов, а в группе *Menu.CATEGORY\_SECONDARY* - три. Причем id элементов и порядок сортировки от группы не зависит.  
  
Запускаем:  
[](http://pics.livejournal.com/coderka/pic/0002kp28/)  
  
Мы видим 5 добавленных элементов и элемент «More» который мы не добавляли, он появляется автоматически, когда количество элементов меню больше шести. Остальные элементы (в нашем случае это все элементы группы Menu.CATEGORY\_SECONDARY) появляются по нажатию на этот пункт (такие элементы называются расширенным меню)  
  
[](http://pics.livejournal.com/coderka/pic/0002pz7s/)  
  
**Выбор пункта меню**  
При нажатии на любой другой элемент, меню просто закрывается. Для выполнения каких-то действий при выборе пункта меню можно использовать метод *onOptionsItemSelected (MenuItem item)*, которому передается ссылка на выбранный пункт меню.

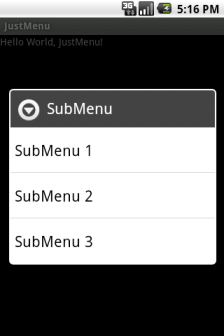
copy to clipboard[подсветка кода](http://aivolkov.ru/online-syntax-highlighter/)

1. @Override
2. **public** **boolean** onOptionsItemSelected (MenuItem item) {
4. **if** (item.getItemId() == 1) {
5. // Элемент «SubMenu», пока ничего не выполняем,
6. // так как на этом элементе позже будет создано подменю
7. }
9. **if** (item.getItemId() == 2) {
10. // меняем значение на противоположенное
11. isGroupEnabled = !isGroupEnabled;
12. // Активируем или отключаем элементы меню в группе Menu.CATEGORY\_SECONDARY
13. myMenu.setGroupEnabled(Menu.CATEGORY\_SECONDARY, isGroupEnabled);
14. // Меняем заголовок тнажатого элемента меню
15. item.setTitle (isGroupEnabled ? "Disable Items" : "Enable Items");
16. }
18. **else** **if** (item.getItemId() == 3) {
19. isGroupVisible = !isGroupVisible;
20. // Показываем или скрываем элементы меню в группе
21. myMenu.setGroupVisible(Menu.CATEGORY\_SECONDARY, isGroupVisible);
22. item.setTitle (isGroupVisible ? "Hide Items" : "Show Items");
23. }
25. **else** **if** (item.getItemId() == 4) {
26. isGroupCheckable = !isGroupCheckable;
27. // Отображаем или скрываем галочку радос с элементами меню
28. myMenu.setGroupCheckable(Menu.CATEGORY\_SECONDARY, isGroupCheckable, **false**);
29. item.setTitle (isGroupCheckable ? "Unset Checkable Items" : "Set Checkable Items");
30. }
32. **else** **if** (item.getItemId() == 5) {
33. isGroupExists = !isGroupExists;
34. **if** (!isGroupExists)
35. // удаляем все элементы меню из группы
36. myMenu.removeGroup(Menu.CATEGORY\_SECONDARY);
37. **else** {
38. // Добавляем элементы меню в группу
39. myMenu.add (Menu.CATEGORY\_SECONDARY, 6, 6, "Item 1");
40. myMenu.add (Menu.CATEGORY\_SECONDARY, 7, 7, "Item 2");
41. myMenu.add (Menu.CATEGORY\_SECONDARY, 8, 8, "Item 3");
43. // Восстанавливаем значения Enable (доступность), Visible (видимость) и Checkable (отображение галочек)
44. myMenu.setGroupEnabled(Menu.CATEGORY\_SECONDARY, isGroupEnabled);
45. myMenu.setGroupVisible(Menu.CATEGORY\_SECONDARY, isGroupVisible);
46. myMenu.setGroupCheckable(Menu.CATEGORY\_SECONDARY, isGroupCheckable, **false**);
47. }
48. item.setTitle (isGroupExists ? "Remove Items" : "Create Items");
49. }
51. **else** **if** (item.getItemId() == 7 || item.getItemId() == 8 || item.getItemId() == 9) {
52. **if** (item.isCheckable())
53. item.setChecked(!item.isChecked());
54. }
55. **return** **true**;
56. }

В метод *boolean onOptionsItemSelected (MenuItem item)* передается выбранный элемент меню — item.   
Метод *item.getItemId()* - Возвращает id элемента, в зависимости от него мы выполняем соответствующие действия.  
  
**Методы для работой с группами элементов меню:**  
*void setGroupEnabled (int group, boolean enabled)* — Устанавливает доступность элементов группы, где *group* — id группы, *enabled* — true(доступны) или false (не доступны)  
  
*void setGroupVisible (int group, boolean visible)* — Устанавливает видимость элементов  
  
*void setGroupCheckable (int group, boolean checkable, boolean exclusive)* — Отображение галочек рядом с элементами, где *exclusive* — исключающая отметка, при exclusive = true, можно выбрать только один из элементов группы  
  
*void removeGroup (int groupId)* — удаление группы элементов  
  
  
**Подменю**  
Теперь сделаем подменю у первого элемента нашего меню (с названием Submenu). Для того чтобы добавить подменю используется метод *SubMenu addSubMenu(int groupId, int itemId, int order, CharSequence title)* класса Menu. Этот метод очень похож на метод add. Метод возвращает объект класса SubMenu, который является производным от Menu, поэтому ему можно добавить элементы по тому же принципу, что и к обычному Menu (с помощью метода add). Добавить подменю к подменю не получиться (компилятор не выдаст ошибки, но во время выполнения программа выдаст исключение).  
  
В методе *onCreateOptionsMenu* заменим строчку

copy to clipboard[подсветка кода](http://aivolkov.ru/online-syntax-highlighter/)

1. // menu.add (Menu.FIRST, 1, 1, "SubMenu");
3. SubMenu sm = menu.addSubMenu (Menu.FIRST, 1, 1, "SubMenu");
4. sm.add(Menu.FIRST, 100, 100, "SubMenu 1");
5. sm.add(Menu.FIRST, 101, 101, "SubMenu 2");
6. sm.add(Menu.FIRST, 102, 102, "SubMenu 3");

Теперь при нажатии на элемент SubMenu откроется новое меню.  
[](http://pics.livejournal.com/coderka/pic/0002q4ek/)  
  
Обработка нажатия на элементы подменю, происходит также в методе *onOptionsItemSelected*.  
  
Во второй части рассмотрим меню с иконками, контекстное меню, а также создание меню в xml. Там же можно скачать исходники.

**Иконки в меню**  
Android поддерживает в меню не только текст, но и изображения. Однако, иконки можно добавить только в обычное меню, ни в подменю, ни в расширенном, ни в контекстном отображаться они не будут (хотя и ошибки не вызовут).  
Добавить иконку можно с помощью метода *setIcon (int resourceId)* класса MenuItem.  
Например так

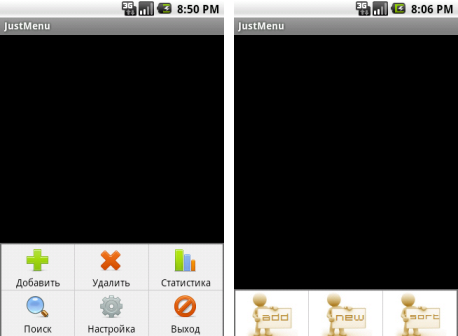
copy to clipboard[подсветка кода](http://aivolkov.ru/online-syntax-highlighter/)

1. MenuItem item = menu.add (Menu.FIRST, 1, 1, "SubMenu");
2. item.setIcon(R.drawable.submenu);

или более коротко

copy to clipboard[подсветка кода](http://aivolkov.ru/online-syntax-highlighter/)

1. menu.add (Menu.FIRST, 1, 1, "SubMenu").setIcon(R.drawable.submenu);

Изображения можно использовать как вместе с текстом, так и отдельно. Например так:  
[](http://pics.livejournal.com/coderka/pic/0002r8yp/)  
  
  
**Контекстное меню**  
Контекстное меню — меню вызываемое при длительном нажатие на какой-то элемент интерфейса (View).   
Для этого надо зарегистрировать View, к которому будет относиться контекстное меню. При создании проекта android в eclipse создается деятельность (Activity) с одним элементом TextView. Привяжем к нему контекстное меню

copy to clipboard[подсветка кода](http://aivolkov.ru/online-syntax-highlighter/)

1. **public** **class** JustMenu **extends** Activity {
3. TextView label;
4. Menu myMenu;
6. ...
8. @Override
9. **public** **void** onCreate(Bundle savedInstanceState) {
10. **super**.onCreate(savedInstanceState);
11. setContentView(R.layout.main);
13. label = (TextView)findViewById(R.id.label);
14. // Регистрируем label для вызова контекстного меню
15. registerForContextMenu(label);
16. }
17. ...
18. }

Для создания контекстного меню используется метод *void onCreateContextMenu (ContextMenu menu, View v, ContextMenuInfo menuInfo)*  
где *menu* — готовое меню (аналогично как в методе onCreateOptionsMenu())  
*v* — View, который сгенерировал вызов контекстного меню  
*menuInfo* — это один из способов, которым View может передать дополнительную информацию в контекстное меню. В простых приложениях его можно просто игнорировать.

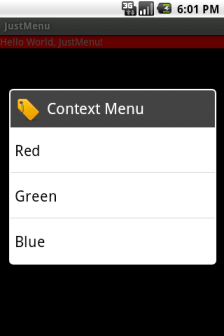
copy to clipboard[подсветка кода](http://aivolkov.ru/online-syntax-highlighter/)

1. @Override
2. **public** **void** onCreateContextMenu (ContextMenu menu, View v, ContextMenuInfo menuInfo) {
3. menu.setHeaderTitle("Context Menu");
4. menu.setHeaderIcon(R.drawable.context);
6. menu.add(Menu.FIRST , 200, 200, "Red");
7. menu.add(Menu.FIRST, 201, 201, "Green");
8. menu.add(Menu.FIRST, 202, 202, "Blue");
9. }

Для обработки нажатия на пункт меню используется метод *boolean onContextItemSelected (MenuItem item)*, работа с которым аналагична onOptionsItemSelected

copy to clipboard[подсветка кода](http://aivolkov.ru/online-syntax-highlighter/)

1. @Override
2. **public** **boolean** onContextItemSelected (MenuItem item) {
3. **if** (item.getItemId() == 200) {
4. label.setBackgroundColor(Color.RED);
5. }
6. **else** **if** (item.getItemId() == 201) {
7. label.setBackgroundColor(Color.GREEN);
8. }
9. **else** **if** (item.getItemId() == 202) {
10. label.setBackgroundColor(Color.BLUE);
11. }
12. **return** **true**;
13. }

В соответствии с выбранным пунктом контекстного меню, меняем фон у текстового элемента.  
[](http://pics.livejournal.com/coderka/pic/0002s599/)  
  
  
**XML разметка меню**  
Меню также как и разметку деятельности (Activity) можно создать в xml. Такие файлы помещаются в /res/menu и для них автоматически генерируется ID ресурса, например R.menu.my\_menu  
Наше меню можно описать в xml так:

copy to clipboard[подсветка кода](http://aivolkov.ru/online-syntax-highlighter/)

1. <?xml version="1.0" encoding="utf-8"?>
2. <menu xmlns:android="http://schemas.android.com/apk/res/android">
3. <group android:id="@+id/groupFirst">
4. <item android:id="@+id/submenu"
5. android:icon="@drawable/submenu"
6. android:title="Submenu" >
7. <menu>
8. <item android:id="@+id/submenu1"
9. android:title="SubMenu 1" />
10. <item android:id="@+id/submenu2"
11. android:title="SubMenu 2" />
12. <item android:id="@+id/submenu3"
13. android:title="SubMenu 3" />
14. </menu>
15. </item>
16. <item android:id="@+id/disable"
17. android:title="Disable Items" />
18. <item android:id="@+id/hide"
19. android:title="Hide Items" />
20. <item android:id="@+id/checkable"
21. android:title="Set Checkable Items" />
22. <item android:id="@+id/remove"
23. android:title="Remove Items" />
24. </group>
25. <group android:id="@+id/groupSecondary">
26. <item android:id="@+id/item1"
27. android:title="Item 1" />
28. <item android:id="@+id/item2"
29. android:title="Item 2" />
30. <item android:id="@+id/item3"
31. android:title="Item 3" />
32. </group>
33. </menu>

Автоматически генерируются id для каждого элемента и группы, например *R.id.submenu* или *R.id.groupSecondary*  
  
Что бы загрузить xml разметку меню используется класс [MenuInflater](http://developer.android.com/reference/android/view/MenuInflater.html" \t "_self). Вначале получаем экземпляр класса*MenuInflater* от текущей деятельности (Activity), а затем с помощью метода *inflate(int resourceId, Menu menu)*загружаем разметку в меню  
  
Таким образом метод onCreateOptionsMenu сокращается:

copy to clipboard[подсветка кода](http://aivolkov.ru/online-syntax-highlighter/)

1. **public** **boolean** onCreateOptionsMenu(Menu menu) {
3. myMenu = menu;
5. MenuInflater mInflater = getMenuInflater();
6. mInflater.inflate(R.menu.my\_menu, menu);
7. **return** **true**;
8. }

Затем в методе *onOptionsItemSelect()* используем сгенерированные id

copy to clipboard[подсветка кода](http://aivolkov.ru/online-syntax-highlighter/)

1. **public** **boolean** onOptionsItemSelected (MenuItem item) {
3. // вместо if (item.getItemId() == 1)
4. **if** (item.getItemId() == R.id.submenu) {
5. // Submenu
6. }
8. // вместо if (item.getItemId() == 2)
9. **if** (item.getItemId() == R.id.disable) {
10. isGroupEnabled = !isGroupEnabled;
11. // вместо myMenu.setGroupEnabled(Menu.CATEGORY\_SECONDARY, isGroupEnabled);
12. myMenu.setGroupEnabled(R.id.groupSecondary, isGroupEnabled);
13. item.setTitle (isGroupEnabled ? "Disable Items" : "Enable Items");
14. }
16. // и т. д.
17. ...
18. }