Меню

Меню являются стандартным компонентом пользовательского интерфейса в приложениях многих типов. Для обеспечения привычной и единообразной технологии работы с приложением следует представлять действия пользователя и другие варианты выбора в своих операциях с помощью API-интерфейсов класса [Menu](https://developer.android.com/reference/android/view/Menu.html?hl=ru).

Начиная с версии Android 3.0 (уровень API 11) в устройствах, работающих под управлением Android, наличие отдельной кнопки *Меню* больше не требуется. С учетом этого изменения приложения для Android должны перестать зависеть от традиционной панели меню из 6 пунктов. Вместо нее в них должна быть строка действий с часто используемыми действиями пользователя.

Несмотря на то что оформление и поведение некоторых пунктов меню изменились, семантика для определения набора действий и вариантов по-прежнему основана на API-интерфейсах класса [Menu](https://developer.android.com/reference/android/view/Menu.html?hl=ru). В этом руководстве рассказывается, как создавать три основополагающих типа меню или представлений действий в системе Android всех версий:

**Меню параметров и строка действий**

Пункты [меню параметров](https://developer.android.com/guide/topics/ui/menus?hl=ru#options-menu) представляют собой основные варианты выбора действий в пределах операции. Именно здесь следует размещать действия, которые затрагивают приложение в целом, например: "Поиск", "Составить сообщение эл. почты" и "Настройки".

При разработке приложений для версии Android 2.3 или более ранних версий пользователи могут открыть панель меню параметров нажатием кнопки *Меню*.

В версии Android 3.0 и последующих версиях пункты меню параметров размещаются в [строке действий](https://developer.android.com/guide/topics/ui/actionbar.html?hl=ru) в виде сочетания отображаемых на экране вариантов действий и раскрывающегося списка дополнительных вариантов выбора. Начиная с Android 3.0 кнопка *Меню* больше не используется (на некоторых устройствах ее нет), поэтому для предоставления доступа к действиям и другим вариантам выбора вам следует перейти к использованию строки действий.

См. раздел [Создание меню параметров](https://developer.android.com/guide/topics/ui/menus?hl=ru#options-menu)

**Контекстное меню и режим контекстных действий**

Контекстное меню ― это [плавающее меню](https://developer.android.com/guide/topics/ui/menus?hl=ru#FloatingContextMenu), которое открывается, когда пользователь длительно нажимает на элемент. В нем содержатся действия, которые затрагивают выбранный контент или контекстный кадр.

При разработке приложения для версии Android 3.0 или выше вместо этого для обеспечения действий с выбранным контентом следует использовать [режим контекстных действий](https://developer.android.com/guide/topics/ui/menus?hl=ru#CAB). В этом режиме в строке, расположенной вверху экрана, отображаются пункты действий, затрагивающие выбранный контент, причем пользователь может выбрать сразу несколько элементов.

См. раздел [Создание контекстного меню](https://developer.android.com/guide/topics/ui/menus?hl=ru#context-menu)

**Всплывающее меню**

Во всплывающем меню отображается вертикальный список пунктов, который привязан к представлению, вызвавшему меню. Он хорошо подходит для предоставления возможности дополнительных вариантов действий, относящихся к определенному контенту или для выдачи вариантов для второй части команды. Действия во всплывающем меню **не** должны напрямую затрагивать соответствующий контент — для этого предназначены контекстные действия. Всплывающее меню предназначено для расширенных действий, относящихся к областям контента в вашей операции.

См. раздел [Создание всплывающего меню](https://developer.android.com/guide/topics/ui/menus?hl=ru#PopupMenu)

Определение меню в файле XML

Для определения пунктов меню всех типов в Android используется стандартный формат XML. Вместо того чтобы создавать меню в коде своей операции, определять меню и все его пункты следует в [ресурсе меню](https://developer.android.com/guide/topics/resources/menu-resource.html?hl=ru)формата XML. После этого ресурс меню можно будет загружать как объект [Menu](https://developer.android.com/reference/android/view/Menu.html?hl=ru) в свои операции или фрагменты.

Использовать ресурсы меню рекомендуется по нескольким причинам:

* в XML проще визуализировать структуру меню;
* это позволяет отделить контент для меню от кода, определяющего работу приложения;
* это позволяет создавать альтернативные варианты меню для разных версий платформы, размеров экрана и других конфигураций путем использования структуры [ресурсов приложения](https://developer.android.com/guide/topics/resources/index.html?hl=ru).

Чтобы определить меню, создайте файл XML в папке res/menu/ вашего проекта и постройте меню со следующими элементами:

<menu>

Определяет класс [Menu](https://developer.android.com/reference/android/view/Menu.html?hl=ru), который является контейнером для пунктов меню. Элемент <menu> должен быть корневым узлом файла, в котором может находиться один или несколько элементов <item> и <group>.

<item>

Создает класс [MenuItem](https://developer.android.com/reference/android/view/MenuItem.html?hl=ru), который представляет один пункт меню. Этот элемент может содержать вложенный элемент <menu> для создания вложенных меню.

<group>

Необязательный, невидимый контейнер для элементов &lt;item&gt;. Он позволяет разделять пункты меню на категории и назначать им одинаковые свойства, такие как активное состояние и видимость. Подробные сведения изложены в разделе [Создание групп меню](https://developer.android.com/guide/topics/ui/menus?hl=ru#groups).

Вот пример меню с именем game\_menu.xml:

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android">  
    <item android:id="@+id/new\_game"  
          android:icon="@drawable/ic\_new\_game"  
          android:title="@string/new\_game"  
          android:showAsAction="ifRoom"/>  
    <item android:id="@+id/help"  
          android:icon="@drawable/ic\_help"  
          android:title="@string/help" />  
</menu>

Элемент <item> поддерживает несколько атрибутов, с помощью которых можно определить внешний вид и поведение пункта меню. Пункты приведенного выше меню имеют следующие атрибуты:

android:id

Идентификатор ресурса, который является уникальным для этого пункта, что позволяет приложению распознавать пункт, когда его выбирает пользователь.

android:icon

Ссылка на графический элемент, который будет использоваться в качестве значка пункта меню.

android:title

Ссылка на строку, которая будет использоваться в качестве названия пункта меню.

android:showAsAction

Указывает, когда и как этот пункт должен отображаться в [строке действий](https://developer.android.com/guide/topics/ui/actionbar.html?hl=ru).

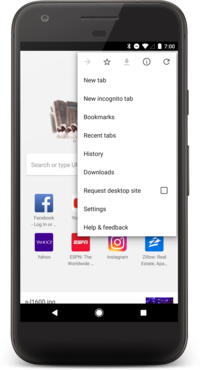
Это самые важные атрибуты, которые следует использовать, но есть также множество других атрибутов. Сведения обо всех поддерживаемых атрибутах см. в документе [Ресурс меню](https://developer.android.com/guide/topics/resources/menu-resource.html?hl=ru).

К пункту любого меню (кроме вложенного меню) можно прикрепить вложенное меню, добавив элемент &lt;menu&gt; в качестве дочернего элемента &lt;item&gt;. Вложенные меню полезны, когда в приложении имеется множество функций, которые можно разделить на категории подобно строке меню приложения для ПК ("Файл", "Правка", "Вид" и т. д.). Например:

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android">  
    <item android:id="@+id/file"  
          android:title="@string/file" >  
        <!-- "file" submenu -->  
        <menu>  
            <item android:id="@+id/create\_new"  
                  android:title="@string/create\_new" />  
            <item android:id="@+id/open"  
                  android:title="@string/open" />  
        </menu>  
    </item>  
</menu>

Для использования меню в операции необходимо загрузить ресурс меню (преобразовать ресурс XML в программируемый объект) с помощью метода [MenuInflater.inflate()](https://developer.android.com/reference/android/view/MenuInflater.html?hl=ru" \l "inflate(int,%20android.view.Menu)). В приведенных далее разделах рассказывается, как загружать меню каждого типа.

Создание меню параметров



**Рисунок 1.** Меню параметров в браузере на Android 2.3.

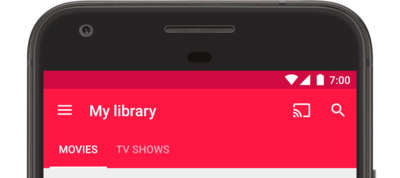
В меню параметров следует размещать действия и другие варианты выбора, которые имеют отношение к контексту текущей операции, например: "Поиск", "Составить сообщение эл. почты" и "Настройки".

Место, где отображаются на экране пункты вашего меню параметров, определяется версией платформы, для которой разработано приложение.

* Если приложение написано для версии **Android 2.3.x (уровень API 10) или более ранней**, содержимое вашего меню параметров отображается внизу экрана, когда пользователь нажимает кнопку *Меню*, как показано на рисунке 1. Когда меню открывается, первой видимой частью является меню значков, в котором имеется шесть пунктов. Если в вашем меню больше шести пунктов, система Android разместит шестой и остальные пункты в дополнительном меню, которое пользователь может открыть, выбрав вариант *Еще*.
* Если приложение предназначено для версии **Android 3.0 (уровень API 11) и более поздних**, пункты меню параметров будут отображаться в [строке действий](https://developer.android.com/guide/topics/ui/actionbar.html?hl=ru). По умолчанию система размещает все действия на панели дополнительных вариантов, которые пользователь может открыть с помощью значка дополнительных действий, расположенного с правой стороны строки действий (либо нажатием кнопки *Меню*, если на устройстве есть такая кнопка). Чтобы обеспечить быстрый доступ к важным действиям, можно принудительно разместить несколько пунктов меню в строке действий, добавивandroid:showAsAction="ifRoom" к соответствующим элементам &lt;item&gt; (см. рисунок 2).

Подробные сведения о пунктах действий и других особенностях строки действий см. в руководстве [Строка действий](https://developer.android.com/guide/topics/ui/actionbar.html?hl=ru).

**Примечание.** Даже если ваше приложение *не* предназначено для работы в версии Android 3.0 или более поздней, можно создать собственный макет со строкой действий, чтобы реализовать похожий эффект. Пример того, как можно поддерживать работу строки действий в старых версиях Android см. в образце кода, иллюстрирующем [Совместимость строки действий](https://developer.android.com/resources/samples/ActionBarCompat/index.html?hl=ru).



**Рисунок 2.** Строка действий из приложения [Honeycomb Gallery](https://developer.android.com/resources/samples/HoneycombGallery/index.html?hl=ru), содержащая вкладки навигации и пункт включения камеры (а также кнопку открытия дополнительной панели действий).

Объявлять пункты меню параметров можно либо из подкласса [Activity](https://developer.android.com/reference/android/app/Activity.html?hl=ru), либо из подкласса [Fragment](https://developer.android.com/reference/android/app/Fragment.html?hl=ru). Если и ваша операция, и фрагменты объявляют пункты меню параметров, в пользовательском интерфейсе они объединяются. Сначала отображаются пункты операции, а за ними следуют пункты фрагментов в том порядке, в котором каждый фрагмент добавляется в операцию. При необходимости можно изменить порядок следования пунктов меню с помощью атрибута android:orderInCategory, указываемого в каждом &lt;item&gt;, который требуется переместить.

Чтобы указать меню параметров для операции, переопределите [onCreateOptionsMenu()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "onCreateOptionsMenu(android.view.Menu)) (фрагменты предоставляют собственный обратный вызов [onCreateOptionsMenu()](https://developer.android.com/reference/android/app/Fragment.html?hl=ru" \l "onCreateOptionsMenu(android.view.Menu,%20android.view.MenuInflater))). В этом методе можно загрузить собственный ресурс меню ([определенный в XML](https://developer.android.com/guide/topics/ui/menus?hl=ru#xml)) в класс [Menu](https://developer.android.com/reference/android/view/Menu.html?hl=ru), имеющийся в обратном вызове. Например:

@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    MenuInflater inflater = [**getMenuInflater()**](https://developer.android.com/reference/android/app/Activity.html?hl=ru#getMenuInflater());  
    inflater.inflate(R.menu.game\_menu, menu);  
    return true;  
}

Пункты меню также можно добавлять с помощью [add()](https://developer.android.com/reference/android/view/Menu.html?hl=ru" \l "add(int,%20int,%20int,%20int)), а получать их с помощью [findItem()](https://developer.android.com/reference/android/view/Menu.html?hl=ru" \l "findItem(int)) для пересмотра их свойств с помощью API-интерфейсов [MenuItem](https://developer.android.com/reference/android/view/MenuItem.html?hl=ru).

Если ваше приложение предназначено для версии Android 2.3.x или более ранней, система вызывает метод [onCreateOptionsMenu()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "onCreateOptionsMenu(android.view.Menu)) для создания меню параметров, когда пользователь открывает это меню впервые. Если приложение предназначено для версии Android 3.0 и или более поздней, система вызывает метод[onCreateOptionsMenu()](https://developer.android.com/reference/android/app/Activity.html?hl=ru#onCreateOptionsMenu(android.view.Menu)) при запуске операции, чтобы отобразить пункты в строке действий.

Обработка нажатий

Когда пользователь выбирает пункт меню параметров (в том числе пункты действий из строки действий), система вызывает метод [onOptionsItemSelected()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "onOptionsItemSelected(android.view.MenuItem)) вашей операции. Этот метод передает выбранный класс [MenuItem](https://developer.android.com/reference/android/view/MenuItem.html?hl=ru). Идентифицировать пункт меню можно, вызвав метод [getItemId()](https://developer.android.com/reference/android/view/MenuItem.html?hl=ru" \l "getItemId()), который возвращает уникальный идентификатор пункта меню (определенный атрибутом android:id из ресурса меню или целым числом, переданным методу [add()](https://developer.android.com/reference/android/view/Menu.html?hl=ru" \l "add(int,%20int,%20int,%20int))). Этот идентификатор можно сопоставить с известными пунктами меню, чтобы выполнить соответствующее действие. Например:

@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    // Handle item selection  
    switch (item.getItemId()) {  
        case R.id.new\_game:  
            newGame();  
            return true;  
        case R.id.help:  
            showHelp();  
            return true;  
        default:  
            return super.onOptionsItemSelected(item);  
    }  
}

Когда пункт меню успешно обработан, возвращается true. Если пункт меню не обрабатывается, следует вызвать реализацию суперкласса [onOptionsItemSelected()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "onOptionsItemSelected(android.view.MenuItem)) (реализация по умолчанию возвращает значение false).

Если в вашей операции имеются фрагменты, система сначала вызовет метод [onOptionsItemSelected()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "onOptionsItemSelected(android.view.MenuItem)) для операции, а затем будет вызывать этот метод для каждого фрагмента (в том порядке, в котором они были добавлены), пока он не возвратит значениеtrue или не закончатся фрагменты.

**Совет.** В Android 3.0 появилась возможность определять в XML поведение при нажатии для пунктов меню с помощью атрибута **android:onClick**. Значением этого атрибута должно быть имя метода, определенное операцией с помощью меню. Этот метод должен быть общедоступным и принимать один параметр **[MenuItem](https://developer.android.com/reference/android/view/MenuItem.html?hl=ru)**, — когда система вызывает этот метод, она передает ему выбранный пункт меню. Подробные сведения и пример см. в документе [Ресурс меню](https://developer.android.com/guide/topics/resources/menu-resource.html?hl=ru).

**Совет.** Если в приложении предусмотрено несколько операций и в некоторых из них имеются одинаковые меню параметров, рассмотрите возможность создания операции, которая будет использовать исключительно методы **[onCreateOptionsMenu()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "onCreateOptionsMenu(android.view.Menu))** и **[onOptionsItemSelected()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "onOptionsItemSelected(android.view.MenuItem))**. Затем распространите этот класс на все операции, у которых должно быть одинаковое меню параметров. Таким образом можно управлять одним набором кода для обработки действий меню, а каждый класс-потомок при этом будет наследовать режимы работы меню. Если требуется добавить пункты меню в одну из операций-потомков, переопределите метод **[onCreateOptionsMenu()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "onCreateOptionsMenu(android.view.Menu))** в этой операции. Вызовите метод **super.onCreateOptionsMenu(menu)**, с тем чтобы создать первоначальные пункты меню, а затем добавьте новые пункты меню с помощью метода **[menu.add()](https://developer.android.com/reference/android/view/Menu.html?hl=ru" \l "add(int,%20int,%20int,%20int))**. Также можно переопределять режимы работы суперкласса для отдельных пунктов меню.

Изменение пунктов меню во время выполнения

После того как система вызовет метод [onCreateOptionsMenu()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "onCreateOptionsMenu(android.view.Menu)), она сохранит заполненный вами экземпляр [Menu](https://developer.android.com/reference/android/view/Menu.html?hl=ru) и будет вызывать метод [onCreateOptionsMenu()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "onCreateOptionsMenu(android.view.Menu)) ,только если меню по каким-то причинам станет некорректным. Однако метод [onCreateOptionsMenu()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "onCreateOptionsMenu(android.view.Menu)) следует использовать только для создания начального состояния меню, а не для внесения в него изменений в течение жизненного цикла операции.

Если вам требуется изменять меню параметров в зависимости от событий, которые возникают в течение жизненного цикла операции, сделать это можно в методе[onPrepareOptionsMenu()](https://developer.android.com/reference/android/app/Activity.html?hl=ru#onPrepareOptionsMenu(android.view.Menu)). Этот метод передает объект [Menu](https://developer.android.com/reference/android/view/Menu.html?hl=ru) в том виде, в котором он в данный момент существует. Его-то и можно изменить путем, например, добавления, удаления или отключения пунктов меню. (Фрагменты также предоставляют обратный вызов [onPrepareOptionsMenu()](https://developer.android.com/reference/android/app/Fragment.html?hl=ru" \l "onPrepareOptionsMenu(android.view.Menu)).)

В версии Android 2.3.x или более ранней система вызывает метод [onPrepareOptionsMenu()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "onPrepareOptionsMenu(android.view.Menu)) каждый раз, когда пользователь открывает меню параметров (нажимает кнопку *Меню* ).

В версии Android 3.0 и последующих версиях считается, что меню параметров всегда открыто, когда пункты меню приведены в строке действий. Когда возникает событие и требуется обновить меню, следует вызвать метод [invalidateOptionsMenu()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "invalidateOptionsMenu()), чтобы запросить у системы вызов метода [onPrepareOptionsMenu()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "onPrepareOptionsMenu(android.view.Menu)).

**Примечание.** Никогда не следует изменять пункты меню параметров с учетом класса **[View](https://developer.android.com/reference/android/view/View.html?hl=ru)**, действующего в данный момент. В сенсорном режиме (когда пользователь не использует трекбол или кнопки направления движения) фокус не может переводиться на представления, поэтому никогда не следует использовать фокус в качестве основы для изменения пунктов меню параметров. Если вы желаете предоставить пункты меню, которые зависят от контекста **[View](https://developer.android.com/reference/android/view/View.html?hl=ru)**, используйте [контекстное меню](https://developer.android.com/guide/topics/ui/menus?hl=ru#context-menu).

Создание контекстного меню



**Рисунок 3**. Снимки экрана с плавающим контекстным меню (слева) и строкой контекстных действий (справа).

В контекстном меню содержатся действия, которые затрагивают определенный элемент или контекстный кадр в пользовательском интерфейсе. Контекстное меню можно создать для любого представления, но чаще всего они используются для элементов из [ListView](https://developer.android.com/reference/android/widget/ListView.html?hl=ru), [GridView](https://developer.android.com/reference/android/widget/GridView.html?hl=ru) или других групп представлений, в которых пользователь может выполнять действия непосредственно с каждым элементом.

Существует два способа предоставления возможности контекстных действий:

* В [плавающем контекстном меню](https://developer.android.com/guide/topics/ui/menus?hl=ru#FloatingContextMenu). Меню отображается в виде плавающего списка пунктов меню (наподобие диалогового окна), когда пользователь длительно нажимает на экран (нажимает и удерживает нажатым) в представлении, которое объявляет поддержку контекстного меню. Пользователи могут каждый раз выполнять контекстное действие только с одним элементом.
* В [режиме контекстных действий](https://developer.android.com/guide/topics/ui/menus?hl=ru#CAB). Этот режим является системной реализацией [ActionMode](https://developer.android.com/reference/android/view/ActionMode.html?hl=ru), которая отображает *строку контекстных действий* вверху экрана с пунктами действий, которые затрагивают выбранные элементы. Когда этот режим активен, пользователи могут одновременно выполнять действие с несколькими элементами (если это допускается приложением).

**Примечание.** Режим контекстных действий поддерживается в версии Android 3.0 (уровень API 11) и последующих версиях. Если этот режим предусмотрен, именно его рекомендуется использовать для отображения контекстных действий. Если ваше приложение поддерживает версии ниже 3.0, то для этих устройств следует вернуться к плавающему контекстному меню.

Создание плавающего контекстного меню

Программирование плавающего контекстного меню

1. Зарегистрируйте класс [View](https://developer.android.com/reference/android/view/View.html?hl=ru), с которым следует связать контекстное меню, вызвав метод [registerForContextMenu()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "registerForContextMenu(android.view.View)) и передав ему [View](https://developer.android.com/reference/android/view/View.html?hl=ru).

Если операция использует [ListView](https://developer.android.com/reference/android/widget/ListView.html?hl=ru) или [GridView](https://developer.android.com/reference/android/widget/GridView.html?hl=ru) и требуется, чтобы каждый элемент предоставлял одинаковое контекстное меню, зарегистрируйте все элементы для контекстного меню, передав [ListView](https://developer.android.com/reference/android/widget/ListView.html?hl=ru) или [GridView](https://developer.android.com/reference/android/widget/GridView.html?hl=ru) методу [registerForContextMenu()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "registerForContextMenu(android.view.View)).

1. Реализуйте метод [onCreateContextMenu()](https://developer.android.com/reference/android/view/View.OnCreateContextMenuListener.html?hl=ru" \l "onCreateContextMenu(android.view.ContextMenu,%20android.view.View,%20android.view.ContextMenu.ContextMenuInfo)) в [Activity](https://developer.android.com/reference/android/app/Activity.html?hl=ru) или [Fragment](https://developer.android.com/reference/android/app/Fragment.html?hl=ru).

Когда зарегистрированное представление примет событие длительного нажатия, система вызовет ваш метод [onCreateContextMenu()](https://developer.android.com/reference/android/view/View.OnCreateContextMenuListener.html?hl=ru" \l "onCreateContextMenu(android.view.ContextMenu,%20android.view.View,%20android.view.ContextMenu.ContextMenuInfo)) . Именно здесь определяются пункты меню. Делается это обычно путем загрузки ресурса меню. Например:

@Override  
public void onCreateContextMenu(ContextMenu menu, View v,  
                                ContextMenuInfo menuInfo) {  
    super.onCreateContextMenu(menu, v, menuInfo);  
    MenuInflater inflater = getMenuInflater();  
    inflater.inflate(R.menu.context\_menu, menu);  
}

[MenuInflater](https://developer.android.com/reference/android/view/MenuInflater.html?hl=ru) позволяет загружать контекстное меню из [ресурса меню](https://developer.android.com/guide/topics/resources/menu-resource.html?hl=ru). В число параметров метода обратного вызова входят [View](https://developer.android.com/reference/android/view/View.html?hl=ru), выбранный пользователем, и объект[ContextMenu.ContextMenuInfo](https://developer.android.com/reference/android/view/ContextMenu.ContextMenuInfo.html?hl=ru), который предоставляет дополнительную информацию о выбранном элементе. Если в вашей операции есть несколько представлений и все они предоставляют разные контекстные меню, то с помощью этих параметров можно определять, какое контекстное меню загружать.

1. Реализуйте метод [onContextItemSelected()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "onContextItemSelected(android.view.MenuItem)).

Когда пользователь выбирает пункт меню, система вызывает этот метод, с тем чтобы вы могли выполнить соответствующее действие. Например:

@Override  
public boolean onContextItemSelected(MenuItem item) {  
    AdapterContextMenuInfo info = (AdapterContextMenuInfo) item.getMenuInfo();  
    switch (item.getItemId()) {  
        case R.id.edit:  
            editNote(info.id);  
            return true;  
        case R.id.delete:  
            deleteNote(info.id);  
            return true;  
        default:  
            return super.onContextItemSelected(item);  
    }  
}

Метод [getItemId()](https://developer.android.com/reference/android/view/MenuItem.html?hl=ru" \l "getItemId()) запрашивает идентификатор для выбранного пункта меню. Идентификаторы должны быть назначены каждому пункту меню в файле XML с помощью атрибута android:id, как описано в разделе [Определение меню в файле XML](https://developer.android.com/guide/topics/ui/menus?hl=ru#xml).

Когда пункт меню успешно обработан, возвращается true. Если пункт меню не обрабатывается, следует передать его реализации суперкласса. Если в вашей операции есть фрагменты, то она получит этот обратный вызов первой. Вызывая суперкласс, когда пункт меню не обрабатывается, система передает событие соответствующему методу обратного вызова в каждом фрагменте по одному (в порядке добавления каждого фрагмента), пока не будет возвращено значение true или false. (Реализация[Activity](https://developer.android.com/reference/android/app/Activity.html?hl=ru) и android.app.Fragment по умолчанию возвращает false, поэтому, когда событие не обрабатывается, следует всегда вызывать суперкласс.)

Использование режима контекстных действий

Режим контекстных действий представляет собой системную реализацию класса [ActionMode](https://developer.android.com/reference/android/view/ActionMode.html?hl=ru), которая направляет пользователя на выполнение контекстных действий при взаимодействии с приложением. Когда пользователь использует этот режим, выбирая элемент, вверху экрана открывается *строка контекстных действий*, содержащая действия, которые пользователь может выполнить с выбранными в данный момент элементами. В этом режиме пользователь может выбирать несколько элементов (если это допускается приложением), снимать выделения с элементов и продолжать навигацию в операции (в тех пределах, в которых это поддерживается приложением). Режим контекстных действий отключается, а строка контекстных действий исчезает, когда пользователь снимет выделение со всех элементов, нажмет кнопку НАЗАД или выберет действие *Готово*, расположенное с левой стороны строки.

**Примечание.** Строка контекстных действий не обязательно бывает связана со [строкой действий](https://developer.android.com/guide/topics/ui/actionbar.html?hl=ru). Они работают независимо друг от друга, даже несмотря на то, что визуально строка контекстных действий занимает положение строки действий.

Если вы разрабатываете приложение для версии Android 3.0 (уровень API 11) или последующих версий, то обычно вместо [плавающего контекстного меню](https://developer.android.com/guide/topics/ui/menus?hl=ru#FloatingContextMenu) вам следует использовать контекстные действия.

Для представлений, которые предоставляют возможность контекстные действия, обычно следует вызывать режим контекстных действий при возникновении одного из двух (или сразу обоих) событий:

* пользователь длительно нажимает в представлении;
* пользователь устанавливает флажок или выбирает другой подобный компонент пользовательского интерфейса в представлении.

То, каким образом ваше представление вызывает режим контекстных действий и определяет поведение каждого действия, зависит от вас. Есть два базовых варианта:

* для контекстных действий в отдельных, произвольных представлениях;
* для пакетных контекстных действий с группами элементов в [ListView](https://developer.android.com/reference/android/widget/ListView.html?hl=ru) или [GridView](https://developer.android.com/reference/android/widget/GridView.html?hl=ru) (что позволяет пользователю выбирать по несколько элементов и выполнять действие с ними всеми).

В следующих разделах описывается конфигурация, необходимая для каждого из этих вариантов.

Включение режима контекстных действий для отдельных представлений

Если вам требуется вызывать режим контекстных действий, только когда пользователь выбирает определенные представления, то вам следует:

1. Реализовать интерфейс [ActionMode.Callback](https://developer.android.com/reference/android/view/ActionMode.Callback.html?hl=ru). В его методах обратного вызова вы можете указать действия для строки контекстных действий, реагировать на нажатия пунктов действий и обрабатывать другие события жизненного цикла для режима действий.
2. Вызывайте [startActionMode()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "startActionMode(android.view.ActionMode.Callback)), когда требуется показать строку (например, когда пользователь выполняет длительное нажатие представления).

Например:

1. Реализуйте интерфейс [ActionMode.Callback](https://developer.android.com/reference/android/view/ActionMode.Callback.html?hl=ru):

private ActionMode.Callback mActionModeCallback = new ActionMode.Callback() {  
  
    // Called when the action mode is created; startActionMode() was called  
    @Override  
    public boolean onCreateActionMode(ActionMode mode, Menu menu) {  
        // Inflate a menu resource providing context menu items  
        MenuInflater inflater = mode.getMenuInflater();  
        inflater.inflate(R.menu.context\_menu, menu);  
        return true;  
    }  
  
    // Called each time the action mode is shown. Always called after onCreateActionMode, but  
    // may be called multiple times if the mode is invalidated.  
    @Override  
    public boolean onPrepareActionMode(ActionMode mode, Menu menu) {  
        return false; // Return false if nothing is done  
    }  
  
    // Called when the user selects a contextual menu item  
    @Override  
    public boolean onActionItemClicked(ActionMode mode, MenuItem item) {  
        switch (item.getItemId()) {  
            case R.id.menu\_share:  
                shareCurrentItem();  
                mode.finish(); // Action picked, so close the CAB  
                return true;  
            default:  
                return false;  
        }  
    }  
  
    // Called when the user exits the action mode  
    @Override  
    public void onDestroyActionMode(ActionMode mode) {  
        mActionMode = null;  
    }  
};

Обратите внимание, что эти обратные вызовы событий почти точно такие же, как и обратные вызовы для [меню параметров](https://developer.android.com/guide/topics/ui/menus?hl=ru#options-menu). Отличаются они только тем, что каждый из них также передает объект [ActionMode](https://developer.android.com/reference/android/view/ActionMode.html?hl=ru), связанный с событием. С помощью API-интерфейсов [ActionMode](https://developer.android.com/reference/android/view/ActionMode.html?hl=ru) можно вносить различные изменения в CAB, например, указывать другой заголовок и подзаголовок с помощью [setTitle()](https://developer.android.com/reference/android/view/ActionMode.html?hl=ru" \l "setTitle(int)) и [setSubtitle()](https://developer.android.com/reference/android/view/ActionMode.html?hl=ru" \l "setSubtitle(int))(удобно для указания количества выбранных элементов).

Также обратите внимание, что приведенный выше образец кода задает для переменной mActionModeзначение null, когда режим действия прекращает свое существование. Далее вы узнаете, каким образом он инициализируется и чем может быть полезно сохранение составной переменной в операции или фрагменте.

1. Для включения режима контекстных действий, когда это необходимо, например, в ответ на длительное нажатие [View](https://developer.android.com/reference/android/view/View.html?hl=ru), вызывайте [startActionMode()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "startActionMode(android.view.ActionMode.Callback)):

someView.setOnLongClickListener(new View.OnLongClickListener() {  
    // Called when the user long-clicks on someView  
    public boolean onLongClick(View view) {  
        if (mActionMode != null) {  
            return false;  
        }  
  
        // Start the CAB using the ActionMode.Callback defined above  
        mActionMode = getActivity().startActionMode(mActionModeCallback);  
        view.setSelected(true);  
        return true;  
    }  
});

При вызове метода [startActionMode()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "startActionMode(android.view.ActionMode.Callback)) система возвращает созданный класс [ActionMode](https://developer.android.com/reference/android/view/ActionMode.html?hl=ru). Сохранив его в составной переменной, вы сможете вносить изменения в строку контекстных действий в ответ на другие события. В приведенном выше образце кода [ActionMode](https://developer.android.com/reference/android/view/ActionMode.html?hl=ru) используется для того, чтобы экземпляр [ActionMode](https://developer.android.com/reference/android/view/ActionMode.html?hl=ru) не создавался повторно, если он уже активен. Достигается это путем проверки, имеет ли элемент значение null перед запуском режима действий.

Включение пакетных контекстных действий в ListView или GridView

Если при наличии набора элементов в [ListView](https://developer.android.com/reference/android/widget/ListView.html?hl=ru) или [GridView](https://developer.android.com/reference/android/widget/GridView.html?hl=ru) (либо другом расширении [AbsListView](https://developer.android.com/reference/android/widget/AbsListView.html?hl=ru)) требуется разрешить пользователям выполнять пакетные действия, следует:

* реализовать интерфейс [AbsListView.MultiChoiceModeListener](https://developer.android.com/reference/android/widget/AbsListView.MultiChoiceModeListener.html?hl=ru) и задать его для группы представлений с помощью метода [setMultiChoiceModeListener()](https://developer.android.com/reference/android/widget/AbsListView.html?hl=ru" \l "setMultiChoiceModeListener(android.widget.AbsListView.MultiChoiceModeListener)); в методах обратного вызова приемника событий вы можете указывать действия для строки контекстных действий, реагировать на нажатия пунктов действий и обрабатывать другие обратные вызовы, унаследованные от интерфейса [ActionMode.Callback](https://developer.android.com/reference/android/view/ActionMode.Callback.html?hl=ru);
* вызвать метод [setChoiceMode()](https://developer.android.com/reference/android/widget/AbsListView.html?hl=ru" \l "setChoiceMode(int)) с аргументом [CHOICE\_MODE\_MULTIPLE\_MODAL](https://developer.android.com/reference/android/widget/AbsListView.html?hl=ru#CHOICE_MODE_MULTIPLE_MODAL).

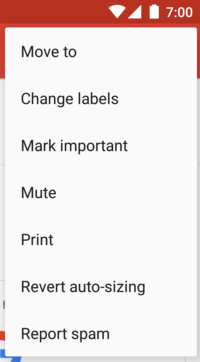
Например:

ListView listView = getListView();  
listView.setChoiceMode(ListView.CHOICE\_MODE\_MULTIPLE\_MODAL);  
listView.setMultiChoiceModeListener(new MultiChoiceModeListener() {  
  
    @Override  
    public void onItemCheckedStateChanged(ActionMode mode, int position,  
                                          long id, boolean checked) {  
        // Here you can do something when items are selected/de-selected,  
        // such as update the title in the CAB  
    }  
  
    @Override  
    public boolean onActionItemClicked(ActionMode mode, MenuItem item) {  
        // Respond to clicks on the actions in the CAB  
        switch (item.getItemId()) {  
            case R.id.menu\_delete:  
                deleteSelectedItems();  
                mode.finish(); // Action picked, so close the CAB  
                return true;  
            default:  
                return false;  
        }  
    }  
  
    @Override  
    public boolean onCreateActionMode(ActionMode mode, Menu menu) {  
        // Inflate the menu for the CAB  
        MenuInflater inflater = mode.getMenuInflater();  
        inflater.inflate(R.menu.context, menu);  
        return true;  
    }  
  
    @Override  
    public void onDestroyActionMode(ActionMode mode) {  
        // Here you can make any necessary updates to the activity when  
        // the CAB is removed. By default, selected items are deselected/unchecked.  
    }  
  
    @Override  
    public boolean onPrepareActionMode(ActionMode mode, Menu menu) {  
        // Here you can perform updates to the CAB due to  
        // an [**invalidate()**](https://developer.android.com/reference/android/view/ActionMode.html?hl=ru#invalidate()) request  
        return false;  
    }  
});

Готово. Теперь, когда пользователь выберет элемент с помощью длительного нажатия, система вызовет метод [onCreateActionMode()](https://developer.android.com/reference/android/view/ActionMode.Callback.html?hl=ru" \l "onCreateActionMode(android.view.ActionMode,%20android.view.Menu)) и отобразит строку контекстных действий с указанными действиями. Пока строка контекстных действий отображается, пользователи могут выбирать дополнительные элементы.

В некоторых случаях, когда контекстные действия содержат общие пункты, можно добавить флажок или другой подобный элемент пользовательского интерфейса, с помощью которого пользователи могут выбирать пункты, поскольку они могут не обнаружить варианта с длительным нажатием. Когда пользователь устанавливает флажок, можно вызывать режим контекстных действий, переводя соответствующий элемент списка в выбранное состояние с помощью [setItemChecked()](https://developer.android.com/reference/android/widget/AbsListView.html?hl=ru" \l "setItemChecked(int,%20boolean)).

Создание всплывающего меню



**Рисунок 4**. Всплывающее меню в приложении Gmail, привязанное к расположенной вверху справа кнопке открытия панели дополнительных пунктов.

[PopupMenu](https://developer.android.com/reference/android/widget/PopupMenu.html?hl=ru) является модальным меню, привязанным к [View](https://developer.android.com/reference/android/view/View.html?hl=ru). Оно отображается ниже представления, к которому привязано, если там есть место, либо поверх него. Варианты использования:

* Предоставление меню с дополнительными пунктами для действий, которые *относятся к* определенному контенту (например, к заголовкам сообщений Gmail, показанным на рисунке 4).

**Примечание.** Оно отличается от контекстного меню, которое обычно используется для действий, *затрагивающих* выбранный контент. Для действий, которые затрагивают выбранный контент, используйте [режим контекстных действий](https://developer.android.com/guide/topics/ui/menus?hl=ru#CAB) или [плавающее контекстное меню](https://developer.android.com/guide/topics/ui/menus?hl=ru#FloatingContextMenu).

* Предоставление второй части командной последовательности (например, кнопки, обозначенной как "Добавить", которая открывает всплывающее меню с различными вариантами добавления);
* Предоставление раскрывающегося меню наподобие [Spinner](https://developer.android.com/reference/android/widget/Spinner.html?hl=ru), которое не сохраняет постоянное выделение.

**Примечание.** Использование класса **[PopupMenu](https://developer.android.com/reference/android/widget/PopupMenu.html?hl=ru)** поддерживается на уровне API 11 и выше.

Если для [определения меню используется XML](https://developer.android.com/guide/topics/ui/menus?hl=ru#xml), вот каким образом можно показать всплывающее меню:

1. Создайте экземпляр класса [PopupMenu](https://developer.android.com/reference/android/widget/PopupMenu.html?hl=ru) с помощью его конструктора, принимающий текущие [Context](https://developer.android.com/reference/android/content/Context.html?hl=ru) и [View](https://developer.android.com/reference/android/view/View.html?hl=ru) приложения, к которым должно быть привязано меню.
2. С помощью [MenuInflater](https://developer.android.com/reference/android/view/MenuInflater.html?hl=ru) загрузите свой ресурс меню в объект [Menu](https://developer.android.com/reference/android/view/Menu.html?hl=ru), возвращенный методом [PopupMenu.getMenu()](https://developer.android.com/reference/android/widget/PopupMenu.html?hl=ru" \l "getMenu()). На API уровня 14 и выше вместо этого можно использовать[PopupMenu.inflate()](https://developer.android.com/reference/android/widget/PopupMenu.html?hl=ru#inflate(int)).
3. Вызовите метод [PopupMenu.show()](https://developer.android.com/reference/android/widget/PopupMenu.html?hl=ru" \l "show()).

Например, вот кнопка с атрибутом [android:onClick](https://developer.android.com/reference/android/R.attr.html?hl=ru" \l "onClick), которая показывает всплывающее меню:

<ImageButton  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:src="@drawable/ic\_overflow\_holo\_dark"  
    android:contentDescription="@string/descr\_overflow\_button"  
    android:onClick="showPopup" />

После этого операция сможет показать вот такое всплывающее меню:

public void showPopup(View v) {  
    PopupMenu popup = new PopupMenu(this, v);  
    MenuInflater inflater = popup.getMenuInflater();  
    inflater.inflate(R.menu.actions, popup.getMenu());  
    popup.show();  
}

На API уровня 14 и выше можно объединить две строки, которые загружают меню, с помощью [PopupMenu.inflate()](https://developer.android.com/reference/android/widget/PopupMenu.html?hl=ru" \l "inflate(int)).

Меню закрывается, когда пользователь выбирает один из пунктов или касается экрана за пределами области меню. Прослушивать событие закрытия меню можно с помощью [PopupMenu.OnDismissListener](https://developer.android.com/reference/android/widget/PopupMenu.OnDismissListener.html?hl=ru).

Обработка нажатий

Для выполнения действия, когда пользователь выбирает пункт меню, необходимо реализовать интерфейс [PopupMenu.OnMenuItemClickListener](https://developer.android.com/reference/android/widget/PopupMenu.OnMenuItemClickListener.html?hl=ru) и зарегистрировать его в своем [PopupMenu](https://developer.android.com/reference/android/widget/PopupMenu.html?hl=ru), вызвав метод [setOnMenuItemclickListener()](https://developer.android.com/reference/android/widget/PopupMenu.html?hl=ru" \l "setOnMenuItemClickListener(android.widget.PopupMenu.OnMenuItemClickListener)). Когда пользователь выбирает пункт меню, система выполняет обратный вызов [onMenuItemClick()](https://developer.android.com/reference/android/widget/PopupMenu.OnMenuItemClickListener.html?hl=ru" \l "onMenuItemClick(android.view.MenuItem)) в вашем интерфейсе.

Например:

public void showMenu(View v) {  
    PopupMenu popup = new PopupMenu(this, v);  
  
    // This activity implements OnMenuItemClickListener  
    popup.setOnMenuItemClickListener(this);  
    popup.inflate(R.menu.actions);  
    popup.show();  
}  
  
@Override  
public boolean onMenuItemClick(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.archive:  
            archive(item);  
            return true;  
        case R.id.delete:  
            delete(item);  
            return true;  
        default:  
            return false;  
    }  
}

Создание групп меню

Группа меню ― это набор пунктов меню с рядом общих характеристик. С помощью группы можно:

* показывать или скрывать все пункты с помощью [setGroupVisible()](https://developer.android.com/reference/android/view/Menu.html?hl=ru" \l "setGroupVisible(int,%20boolean));
* включать или отключать все пункты с помощью [setGroupEnabled()](https://developer.android.com/reference/android/view/Menu.html?hl=ru" \l "setGroupEnabled(int,%20boolean));
* Указывать, можно ли помечать все пункты, с помощью [setGroupCheckable()](https://developer.android.com/reference/android/view/Menu.html?hl=ru" \l "setGroupCheckable(int,%20boolean,%20boolean)).

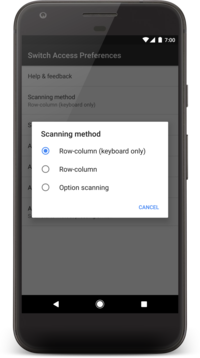
Для создания группы необходимо вложить элементы &lt;item&gt; в элемент &lt;group&gt; в своем ресурсе меню либо указать идентификатор группы с помощью метода [add()](https://developer.android.com/reference/android/view/Menu.html?hl=ru" \l "add(int,%20int,%20int,%20int)).

Вот пример ресурса меню, в котором имеется группа:

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android">  
    <item android:id="@+id/menu\_save"  
          android:icon="@drawable/menu\_save"  
          android:title="@string/menu\_save" />  
    <!-- menu group -->  
    <group android:id="@+id/group\_delete">  
        <item android:id="@+id/menu\_archive"  
              android:title="@string/menu\_archive" />  
        <item android:id="@+id/menu\_delete"  
              android:title="@string/menu\_delete" />  
    </group>  
</menu>

Элементы, находящиеся в группе, отображаются на одном уровне с первым элементом — все три пункта меню являются элементами одного уровня. Однако можно изменить характеристики двух пунктов из группы, указав ссылку на идентификатор группы и воспользовавшись приведенными выше методами. Кроме того, система никогда не будет разделять сгруппированные пункты. Например, если объявить android:showAsAction="ifRoom" для каждого пункта, то они оба будут отображены либо в строке действий, либо в дополнительных действиях.

Использование пунктов меню, которые можно пометить



**Рисунок 5**. Снимок экрана с вложенным меню, пункты которого можно пометить.

Такое меню можно использовать в качестве интерфейса для включения и отключения тех или иных параметров. При этом для автономных параметров используется флажок, а для групп взаимоисключающих вариантов ― переключатель. На рисунке 5 показано вложенное меню с пунктами, которые можно выбирать с помощью переключателей.

**Примечание.** Пункты в меню значков (из меню параметров) не могут отображать флажки или переключатели. Если вы решите сделать так, чтобы пункты меню значков можно было помечать, вам необходимо будет вручную указать помеченное состояние путем замены значка и/или теста при каждом изменении состояния.

Определять возможность помечать отдельные пункты меню можно с помощью атрибута android:checkable в элементе &lt;item&gt;, а для всей группы это делается с помощью атрибута android:checkableBehavior в элементе &lt;group&gt;. Например , все пункты этой группы меню можно помечать с помощью переключателей:

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android">  
    <group android:checkableBehavior="single">  
        <item android:id="@+id/red"  
              android:title="@string/red" />  
        <item android:id="@+id/blue"  
              android:title="@string/blue" />  
    </group>  
</menu>

Атрибут android:checkableBehavior принимает один из трех параметров:

single

Только один пункт из группы можно пометить (переключатель)

all

Все пункты можно пометить (флажки)

none

Пометить нельзя ни один пункт

Для того чтобы применить к пункту помеченное состояние по умолчанию, служит атрибут android:checked в элементе &lt;item&gt;, а изменить его в коде можно с помощью метода [setChecked()](https://developer.android.com/reference/android/view/MenuItem.html?hl=ru" \l "setChecked(boolean)).

Когда выбирается пункт, который может быть помечен, система вызывает соответствующий ему метод обратного вызова (например [onOptionsItemSelected()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "onOptionsItemSelected(android.view.MenuItem))). Именно здесь необходимо задать состояние флажка, поскольку флажок или переключатель не изменяет свое состояние автоматически. Запросить текущее состояние пункта (в котором он находился до того, как был выбран пользователем) можно с помощью[isChecked()](https://developer.android.com/reference/android/view/MenuItem.html?hl=ru#isChecked()), а затем задать помеченное состояние с помощью [setChecked()](https://developer.android.com/reference/android/view/MenuItem.html?hl=ru" \l "setChecked(boolean)). Например:

@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.vibrate:  
        case R.id.dont\_vibrate:  
            if (item.isChecked()) item.setChecked(false);  
            else item.setChecked(true);  
            return true;  
        default:  
            return super.onOptionsItemSelected(item);  
    }  
}

Если помеченное состояние не установить этим способом, то, когда пользователь выберет пункт, его отображаемое состояние (флажок или переключатель) не изменится. Если же помеченное состояние установить, операция сохранит его для пункта, с тем чтобы, когда пользователь откроет это меню, он увидел, что галочка поставлена.

**Примечание.** Пункты меню, которые можно пометить галочкой, предназначены для использования только в рамках одного сеанса. Они не сохраняются после прекращения существования приложения. Если имеются настройки приложения, которые требуется сохранить для пользователя, делать это следует с помощью [общих настроек](https://developer.android.com/guide/topics/data/data-storage.html?hl=ru#pref).

Добавление пунктов меню на основе объектов Intent

Иногда требуется, чтобы пункт меню запускал операцию с помощью объекта [Intent](https://developer.android.com/reference/android/content/Intent.html?hl=ru) (это может быть операция как из вашего, так и из другого приложения). Когда вам известен объект Intent, который требуется использовать, и у вас есть определенный пункт меню, который должен инициировать этот объект Intent, можно выполнить объект Intent с помощью [startActivity()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "startActivity(android.content.Intent)) во время выполнения соответствующего метода обратного вызова, запускаемого при выборе пункта меню (например, обратного вызова [onOptionsItemSelected()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "onOptionsItemSelected(android.view.MenuItem))).

Однако если вы не уверены, что на устройстве пользователя есть приложение, которое может обработать этот объект Intent, добавление пункта меню, который его вызывает, может привести к тому, что он не будет работать, поскольку объект Intent может не быть передан в операцию. Чтобы решить эту проблему, Android позволяет динамически добавлять в меню пункты, когда система Android обнаруживает на устройстве операции, которые могут обработать ваш объект Intent.

Добавление пунктов меню на основе имеющихся операций, которые принимают объект Intent:

1. Определите объект с категорией [CATEGORY\_ALTERNATIVE](https://developer.android.com/reference/android/content/Intent.html?hl=ru#CATEGORY_ALTERNATIVE) и/или [CATEGORY\_SELECTED\_ALTERNATIVE](https://developer.android.com/reference/android/content/Intent.html?hl=ru#CATEGORY_SELECTED_ALTERNATIVE), а также с любыми другими условиями.
2. Вызовите метод [Menu.addIntentOptions()](https://developer.android.com/reference/android/view/Menu.html?hl=ru" \l "addIntentOptions(int,%20int,%20int,%20android.content.ComponentName,%20android.content.Intent[],%20android.content.Intent,%20int,%20android.view.MenuItem[])). Затем Android выполнит поиск приложений, которые могут выполнить этот объект Intent, и добавит их в ваше меню.

При отсутствии установленных приложений, которые удовлетворяют требованиям объекта Intent, ни одного пункта меню добавлено не будет.

**Примечание.** [**CATEGORY\_SELECTED\_ALTERNATIVE**](https://developer.android.com/reference/android/content/Intent.html?hl=ru#CATEGORY_SELECTED_ALTERNATIVE) используется для обработки элемента, выбранного в данный момент на экране. Поэтому его следует использовать только при создании меню в **[onCreateContextMenu()](https://developer.android.com/reference/android/app/Activity.html?hl=ru" \l "onCreateContextMenu(android.view.ContextMenu,%20android.view.View,%20android.view.ContextMenu.ContextMenuInfo))**.

Например:

@Override  
public boolean onCreateOptionsMenu(Menu menu){  
    super.onCreateOptionsMenu(menu);  
  
    // Create an Intent that describes the requirements to fulfill, to be included  
    // in our menu. The offering app must include a category value of Intent.CATEGORY\_ALTERNATIVE.  
    Intent intent = new Intent(null, dataUri);  
    intent.addCategory(Intent.CATEGORY\_ALTERNATIVE);  
  
    // Search and populate the menu with acceptable offering applications.  
    menu.addIntentOptions(  
         R.id.intent\_group,  // Menu group to which new items will be added  
         0,      // Unique item ID (none)  
         0,      // Order for the items (none)  
         this.getComponentName(),   // The current activity name  
         null,   // Specific items to place first (none)  
         intent, // Intent created above that describes our requirements  
         0,      // Additional flags to control items (none)  
         null);  // Array of MenuItems that correlate to specific items (none)  
  
    return true;  
}

Каждая обнаруженная операция, в которой имеется фильтр Intent, соответствующий данному объекту Intent, добавляется в виде пункта меню. Для этого значение из элемента android:label фильтра Intent используется в качестве заголовка пункта меню, а значок приложения ― в качестве значка этого пункта меню. Метод [addIntentOptions()](https://developer.android.com/reference/android/view/Menu.html?hl=ru" \l "addIntentOptions(int,%20int,%20int,%20android.content.ComponentName,%20android.content.Intent[],%20android.content.Intent,%20int,%20android.view.MenuItem[])) возвращает количество добавленных пунктов меню.

**Примечание.** При вызове метода **[addIntentOptions()](https://developer.android.com/reference/android/view/Menu.html?hl=ru" \l "addIntentOptions(int,%20int,%20int,%20android.content.ComponentName,%20android.content.Intent[],%20android.content.Intent,%20int,%20android.view.MenuItem[]))** он переопределяет все пункты меню по группе меню, указанной в первом аргументе.

Предоставление возможности добавить свою операцию в другие меню

Вы также можете предоставить другим приложениям возможность воспользоваться своей операцией, с тем чтобы ваше приложение могло быть включено в меню других приложений (процедура, обратная описанной выше).

Чтобы операция могла быть включена в меню других приложений, необходимо определить фильтр Intent обычным образом, но непременно включить в него значения [CATEGORY\_ALTERNATIVE](https://developer.android.com/reference/android/content/Intent.html?hl=ru#CATEGORY_ALTERNATIVE) и/или [CATEGORY\_SELECTED\_ALTERNATIVE](https://developer.android.com/reference/android/content/Intent.html?hl=ru#CATEGORY_SELECTED_ALTERNATIVE) для категории фильтра Intent. Например:

<intent-filter label="@string/resize\_image">  
    ...  
    <category android:name="android.intent.category.ALTERNATIVE" />  
    <category android:name="android.intent.category.SELECTED\_ALTERNATIVE" />  
    ...  
</intent-filter>

Подробные сведения о написании фильтров Intent см. в документе [Объекты Intent и фильтры объектов Intent](https://developer.android.com/guide/components/intents-filters.html?hl=ru).

Образец приложения, в котором используется эта методика, см. в образце кода [Note Pad](https://developer.android.com/resources/samples/NotePad/src/com/example/android/notepad/NoteEditor.html?hl=ru).