

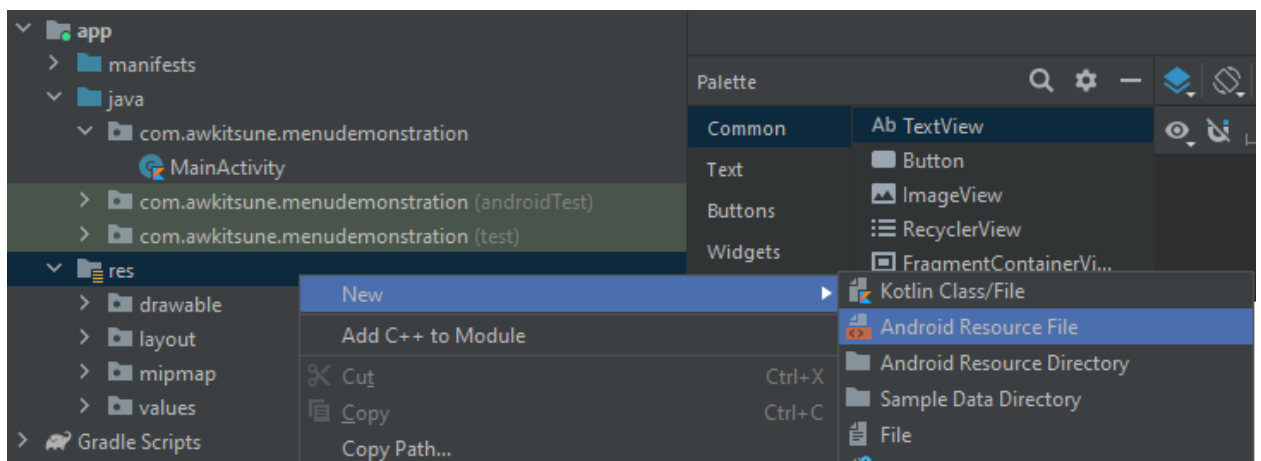


Лекция #24. Меню

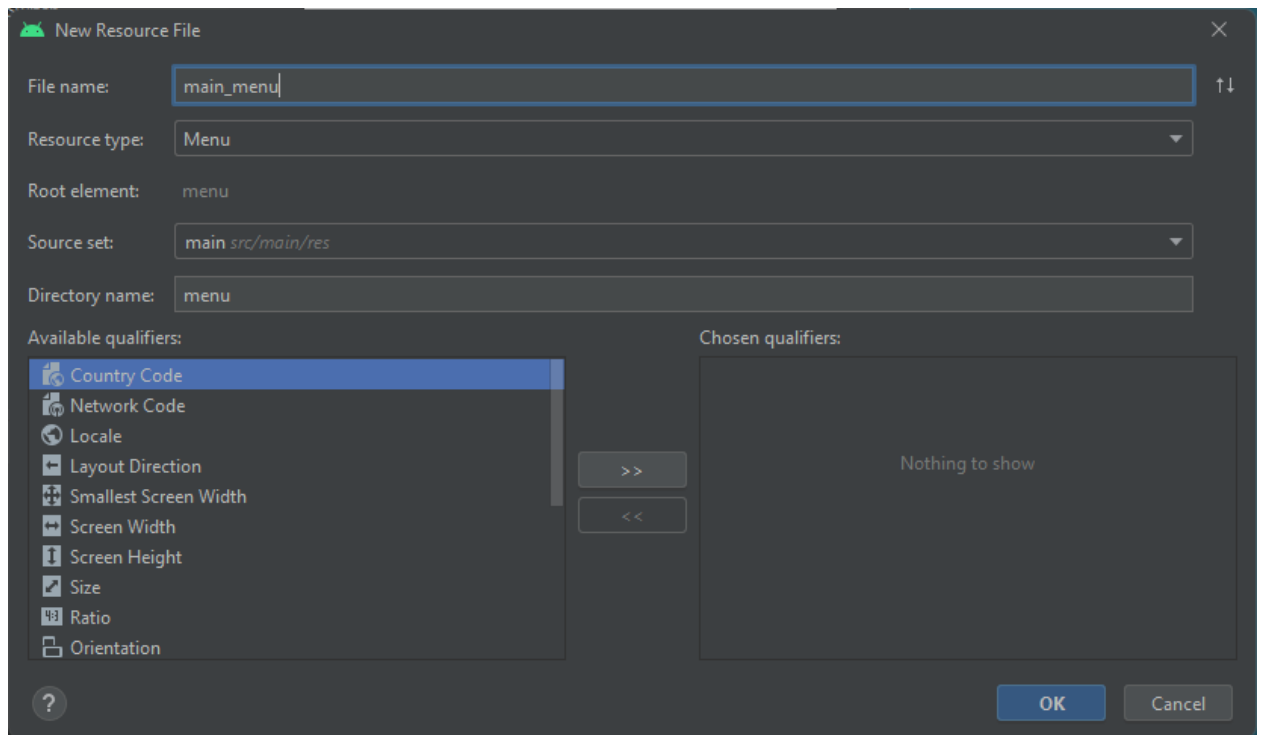
Меню в приложениях представляет класс **android.view.Menu**, и каждая **activity** ассоциируется с объектом этого типа. Объект **android.view.Menu** может включать различное количество элементов, а те в свою очередь могут хранить подэлементы.

Определение меню в xml

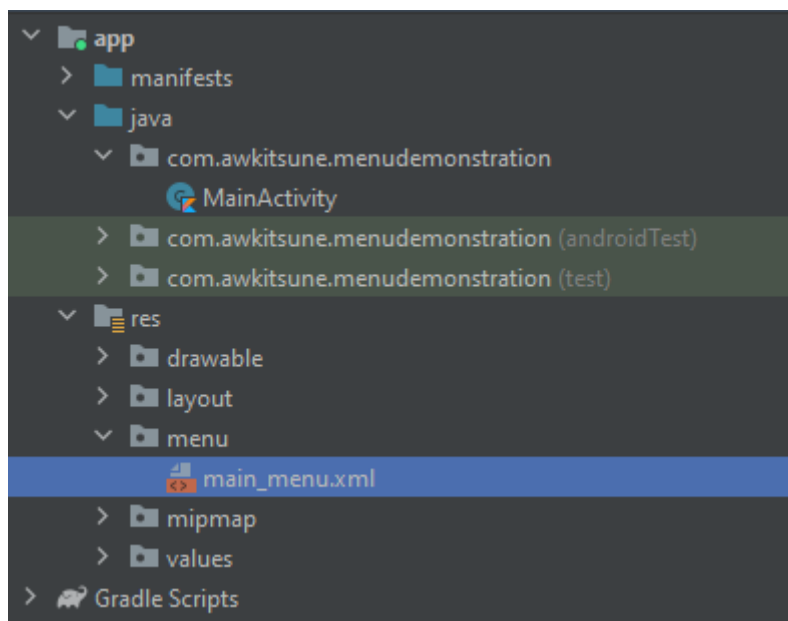
Меню, как и файлы интерфейса или изображений, также представляет собой ресурс. Однако при создании нового проекта с **Empty Activity** по умолчанию нет никаких ресурсов меню, поэтому при необходимости их нужно добавлять вручную. Так, для определения ресурсов меню в проекте нажмем правой кнопкой мыши в проекте на каталог **res** и далее в открывшемся списке выберем пункт **New -> Android Resource File**:



Далее в появившемся окне укажем для имени файла название **main_menu**, а для поля **Resource Type** (тип ресурса) выберем **Menu**:



После этого в каталоге res будет создан подкаталог menu, в котором будет находиться файл **main_menu.xml**.



По умолчанию этот файл определяет один пустой элемент menu:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

</menu>
```

Изменим содержимое файла, определив несколько пунктов:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/account_menu_item"
        android:title="Аккаунт" />
    <item
        android:id="@+id/settings_menu_item"
        android:title="Настройки" />
    <item
        android:id="@+id/about_menu_item"
        android:title="О программе" />
</menu>
```

Тег **<menu>** является корневым узлом файла и определяет меню, состоящее из одного или нескольких элементов **<item>** и **<group>**.

Элемент **<item>** представляет объект **MenuItem**, который является одним из элементов меню. Этот элемент может содержать внутренний подэлемент **<menu>**, с помощью которого создается подменю.

Элемент **<item>** включает следующие атрибуты, которые определяют его внешний вид и поведение:

- **android:id**: уникальный id элемента меню, который позволяет его опознать при выборе пользователем и найти через поиск ресурса по id
- **android:icon**: ссылка на ресурс drawable, который задает изображение для элемента (**android:icon="@drawable/ic_help"**)
- **android:title**: ссылка на ресурс строки, содержащий заголовок элемента. По умолчанию имеет значение **"Settings"**
- **android:orderInCategory**: порядок следования элемента в меню

Наполнение меню элементами

Мы определили меню с тремя элементами, но само определение элементов в файле еще не создает меню. Это всего лишь **декларативное описание**. Чтобы вывести его на экран, нам надо использовать его в классе **Activity**. Для этого надо переопределить метод **onCreateOptionsMenu**. Итак, перейдем к классу **MainActivity** и изменим его следующим образом:

```
package com.awkitsune.menudemonstration

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.Menu

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

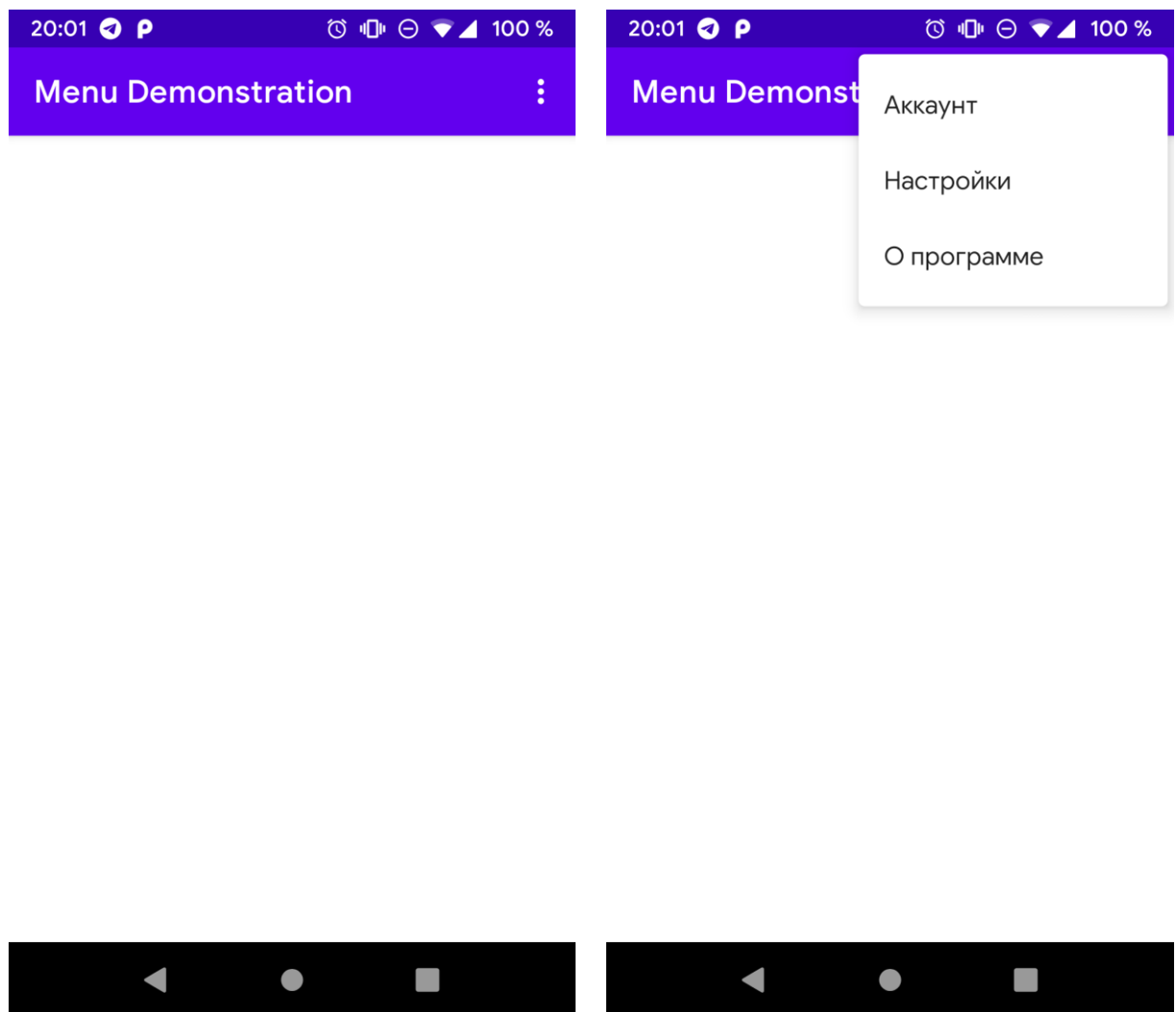
```

override fun onCreateOptionsMenu(menu: Menu?): Boolean {
    menuInflater.inflate(R.menu.main_menu)
    return true
}

```

Свойство **menuInflater** возвращает объект **MenuInflater**, у которого вызывается метод **inflate()**. Этот метод в качестве первого параметра принимает ресурс, представляющий наше декларативное описание меню в **xml**, и наполняет им объект **menu**, переданный в качестве второго параметра.

Запустим приложение по умолчанию и нажмем на кнопку меню в правом верхнем углу:



Обработка нажатий в меню

Если мы нажмем на любой из пунктов меню, то ничего не произойдет. Чтобы привязать к меню действия, нам надо переопределить в классе activity **onOptionsItemSelected**.

Для вывода выбранного элемента меню в файле **activity_main.xml** определим текстовое поле с **id=header**:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"
        android:gravity="center"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

И изменим класс **MainActivity**:

```
package com.awkitsune.menudemonstration

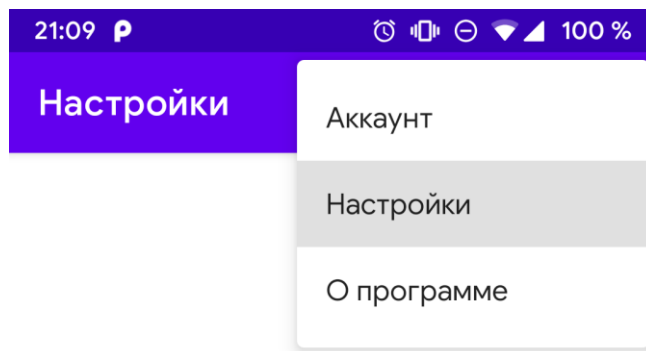
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }

    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.main_menu, menu)
        return true
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        when(item.itemId) {
            R.id.account_menu_item -> title = "Аккаунт"
            R.id.settings_menu_item -> title = "Настройки"
            R.id.about_menu_item -> title = "О программе"
        }
        return super.onOptionsItemSelected(item)
    }
}
```

Чтобы понять, какой пункт меню выбран, вначале получаем его идентификатор **item.itemId** в конструкции **when** и выбираем нужный вариант. В зависимости от выбора производим определенные действия - в данном случае устанавливаем заголовок **activity**.



Hello World!



Стоит отметить, что в данном случае, если наша задача заключалась, чтобы просто в выводе текста выбранного пункта меню, то мы вместо конструкции **switch** просто могли написать так:

```
title = item.title;
```

Программное создание меню

Кроме определения элементов меню в **xml**, можно также создать меню программным способом. Для добавления новых пунктов меню используется метод **add()** класса **Menu**.

Например, изменим код **MainActivity**:

```
package com.awkitsune.menudemonstration

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }

    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        super.onCreateOptionsMenu(menu)

        menu?.add("Аккаунт")
        menu?.add("Настройки")
        menu?.add("О программе")

        return true
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        title = item.title
        return super.onOptionsItemSelected(item)
    }
}
```

Используемая версия метода **add()** принимает заголовок для пункта меню.

Группы в меню и подменю

Создание подменю

Для создания подменю в файле разметки меню определим внутренний элемент **menu**:

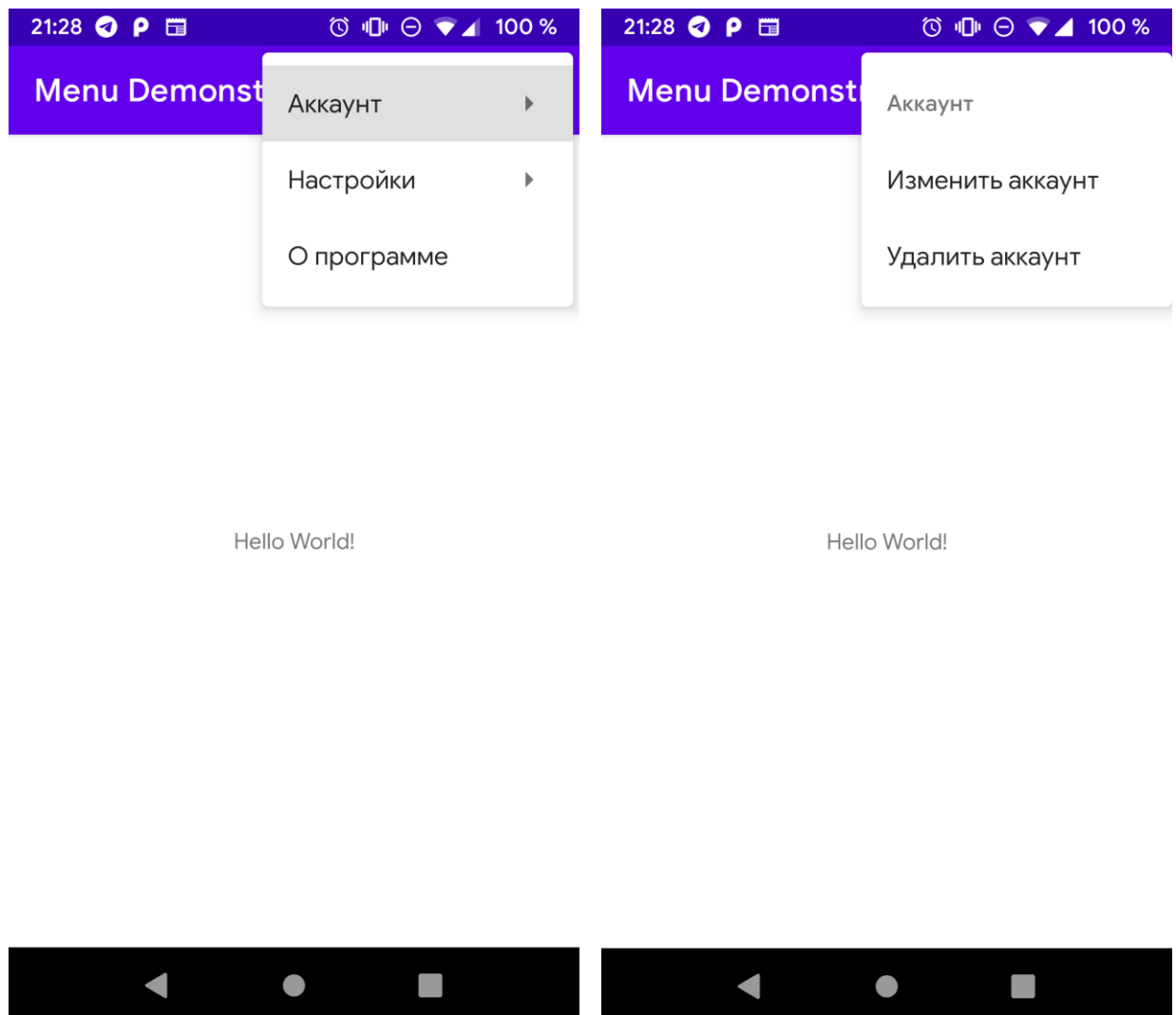
```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/account_menu_item"
        android:title="Аккаунт" >
        <menu>
            <item
                android:id="@+id/edit_account"
                android:title="Изменить аккаунт"/>
        </menu>
    </item>
</menu>
```

```

        <item
            android:id="@+id/remove_account"
            android:title="Удалить аккаунт"/>
    </menu>
</item>
<item
    android:id="@+id/settings_menu_item"
    android:title="Настройки" >
    <menu>
        <item
            android:id="@+id/open_settings"
            android:title="Открыть настройки"/>
        <item
            android:id="@+id/save_settings"
            android:title="Сохранить настройки"/>
    </menu>
</item>
<item
    android:id="@+id/about_menu_item"
    android:title="О программе" >
</item>
</menu>

```

После нажатия на меню отображаются элементы верхнего уровня, по нажатию на которые мы можем перейти к подменю:



Группы в меню

Использование элемента **group** позволяет оформить элементы меню в группу:

```
<?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/account_menu_item"
      android:title="Аккаунт"
      android:checked="true">
    </item>
    <item
      android:id="@+id/settings_menu_item"
      android:title="Настройки"
    </item>
    <item
      android:id="@+id/about_menu_item"
      android:title="О программе"
    </item>
  </group>
</menu>
```

В определении группы мы можем установить атрибут **android:checkableBehavior**. Этот атрибут может принимать следующие значения:

- **single** (у каждого элемента создается радиокнопка)
- **all** (для каждого элемента создается флажок)
- **none**

В данном случае для каждого элемента будет создаваться радиокнопка (**кружок**). И для первого элемента устанавливается отмеченная радиокнопка (**android:checked="true"**).

В файле разметки интерфейса **activity_main.xml** у нас уже определено текстовое поле **textView**

А в классе **MainActivity** определим выделение радиокнопки у выбранного пункта меню:

```
override fun onOptionsItemSelected(item: MenuItem): Boolean {
    val textView: TextView = findViewById(R.id.textView)

    when(item.itemId) {
        R.id.account_menu_item -> {
            textView.text = "Аккаунт"
            item.isChecked = true
            return true
        }

        R.id.settings_menu_item -> {
```

```

        textView.text = "Настройки"
        item.isChecked = true
        return true
    }
    R.id.about_menu_item -> {
        textView.text = "О программе"
        item.isChecked = true
        return true
    }
}
return super.onOptionsItemSelected(item)
}

```

Программное создание групп в меню и подменю

Также группы и подменю можно создавать программным способом. Так, изменим код **MainActivity**:

```

package com.awkitsune.menudemonstration

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import android.widget.TextView

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }

    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        super.onCreateOptionsMenu(menu)
        menu?.add(0, //Группа
            1, //Id
            1, //Порядок
            "Создать") //Заголовок
        menu?.add(0, 2, 1, "Открыть")
        menu?.add(0, 3, 2, "Сохранить")

        return true
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        val textView: TextView = findViewById(R.id.textView)

        when(item.itemId){
            1 -> {
                textView.text = "Создать документ"
                return true
            }
            2 -> {
                textView.text = "Открыть документ"
                return true
            }
            3 -> {
                textView.text = "Сохранить документ"
                return true
            }
        }
    }
}

```

```
    return super.onOptionsItemSelected(item)
}
```

