

### Лабораторная работа №9. Работа с меню

Задание: создать приложение, использующее опциональное и контекстное меню для какого-нибудь интерфейсного элемента. Естественно, выбор пунктов меню должен что-то менять в интерфейсных элементах или их отображении. Например, очистить поле ввода через контекстное меню, или отобразить невидимые интерфейсные элементы через установку галок в опциональном меню.

В операционной системе Android есть два вида меню: контекстное и опциональное. Опциональное меню появляется при нажатии на кнопку меню на устройстве, либо при нажатии кнопки меню в программе (три точки или три тире в правом или левом верхнем углу в зависимости от версии ОС). Контекстное меню появляется при длительном нажатии на каком-либо элементе пользовательского интерфейса.

Создайте новый проект с Empty Activity. Конечно, можно создать проект, выбрав Basic Activity, при этом опциональное меню добавится автоматически: в папке res каждого проекта будет отдельная папка menu, в которой располагается файл menu\_main.xml, описывающий опциональное меню приложения по умолчанию. В то же время, для опционального меню в файле MainActivity.java добавится заготовленный код с двумя методами для опционального меню: onCreateOptionsMenu и onOptionsItemSelected; а в интерфейсе приложения и в коде по умолчанию будет один пункт меню Settings.

Однако наша цель – научиться создавать меню самостоятельно, поэтому всё надо делать вручную – это недолго. Создадим папку menu в папке res (правой кнопкой мыши по папке res -> New -> Directory) и добавим в неё файл menu.xml: это можно сделать, нажав правой кнопкой мыши по папке menu и выбрав New и появившийся пункт Menu resource file. Заметьте, этот пункт контекстного меню папки menu отсутствует в контекстных меню других папок. В открывшемся окне назовите файл также menu, см. рисунок 38.

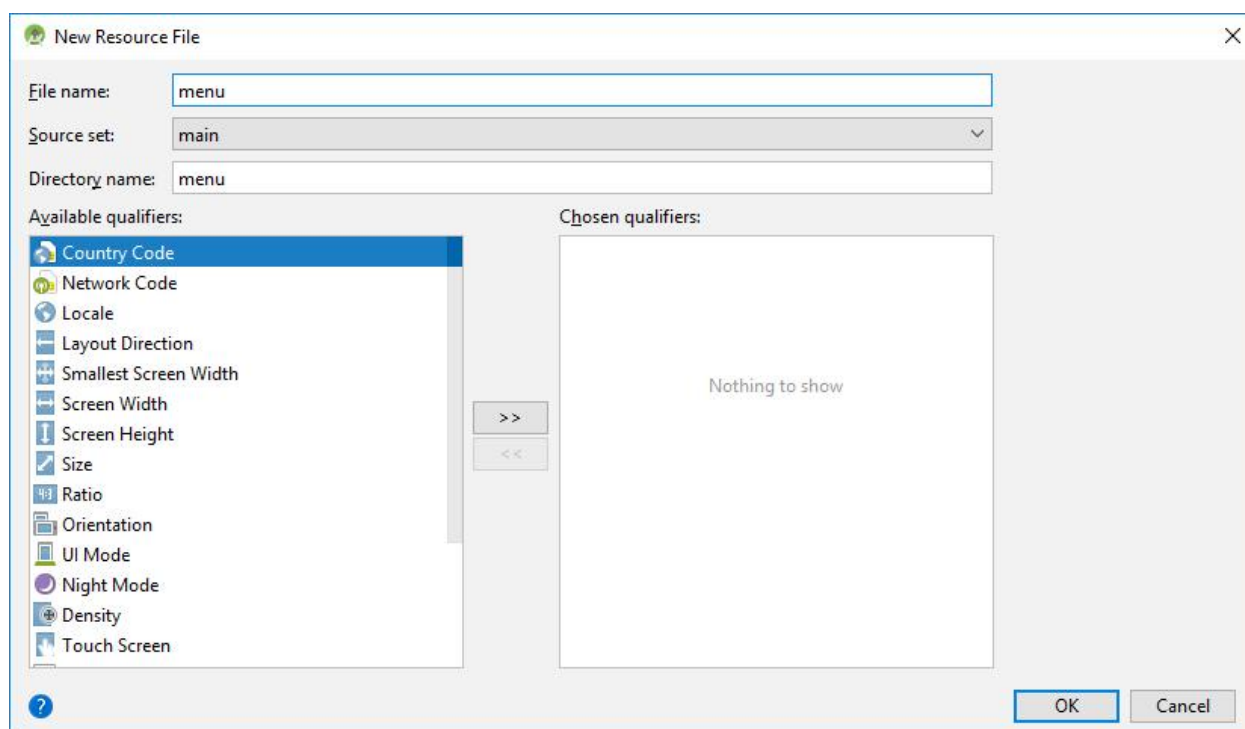


Рисунок 38 – Создание файла опционального меню

Перейдите в файл menu.xml (закладка Text) и добавьте новый пункт опционального меню:

```
<item android:id="@+id/show_text" android:title="Скрыть текст" android:checkable="true" />
```

Теперь необходимо этот пункт меню отобразить в приложении. Перейдите в файл MainActivity.java и добавьте метод onOptionsItemSelected после метода onCreate: начните печатать onOpt... и выберите конструктор для этого метода. В методе onOptionsItemSelected перед return напишите следующее:

```
int id = item.getItemId();
if (id == R.id.show_text){
    if (item.isChecked()){
        TextView textView = findViewById(R.id.text1);
        textView.setVisibility(View.VISIBLE);
        item.setChecked(false);
    }
    else {
        TextView textView = findViewById(R.id.text1);
        textView.setVisibility(View.INVISIBLE);
        item.setChecked(true);
    }
}
```

В данном коде в опциональное меню добавляется новый пункт, который в зависимости от состояния поля checkbox скрывает или показывает элемент TextView. Однако только этого метода onOptionsItemSelected недостаточно, необходимо добавить в MainActivity.java ещё один метод:

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

В этом методе опциональное меню меню заполняется содержимым, которое лежит в файле menu.xml, т.е. – нашим единственным пунктом Скрыть текст. Кроме того, без этого метода меню не будет видно в интерфейсе приложения. Запустите проект и проверьте, что всё работает (рисунок 39).

Теперь создадим контекстное меню, для этого необходимо создать ещё один xml-файл в папке menu. Нажмите по папке menu правой кнопкой мыши, затем New -> Menu resource file. Оставьте все

поля по умолчанию и назовите файл contextmenu. После создания перейдите в данный файл (закладка Text) и напишите:

```
<item android:id="@+id/color_red" android:title="Красный" />
<item android:id="@+id/color_black" android:title="Чёрный" />
```

Вернитесь в файл MainActivity.java и в методе onCreate напишите:

```
TextView textView = findViewById(R.id.text1);
registerForContextMenu(textView);
```

Таким образом, мы зарегистрировали элемент TextView в контекстном меню, и теперь добавленное контекстное меню будет работать для этого текстового элемента. Затем создайте новый метод onCreateContextMenu после метода onCreate. Начните писать onCreateC и выберите шаблон. В методе onCreateContextMenu первой строчкой перед существующим кодом напишите:

```
getMenuInflater().inflate(R.menu.contextmenu, menu);
```

Далее необходимо создать ещё один метод, по аналогии с меню опций: onOptionsItemSelected. Начните писать onContextI и выберите предложенный шаблон. В созданном методе onOptionsItemSelected нужно описать кнопки и действия:

```
int id = item.getItemId();
if (id == R.id.color_red){
    TextView textView = (TextView) findViewById(R.id.text1);
    textView.setTextColor(Color.parseColor("red"));
}
if (id == R.id.color_black){
    TextView textView = (TextView) findViewById(R.id.text1);
    textView.setTextColor(Color.parseColor("black"));
}
```

Контекстное меню создано. Запустите приложение, нажмите и держите кнопку мыши / палец на элементе TextView, после чего появится контекстное меню с двумя созданными пунктами, которые меняют цвет текста (рисунок 40).

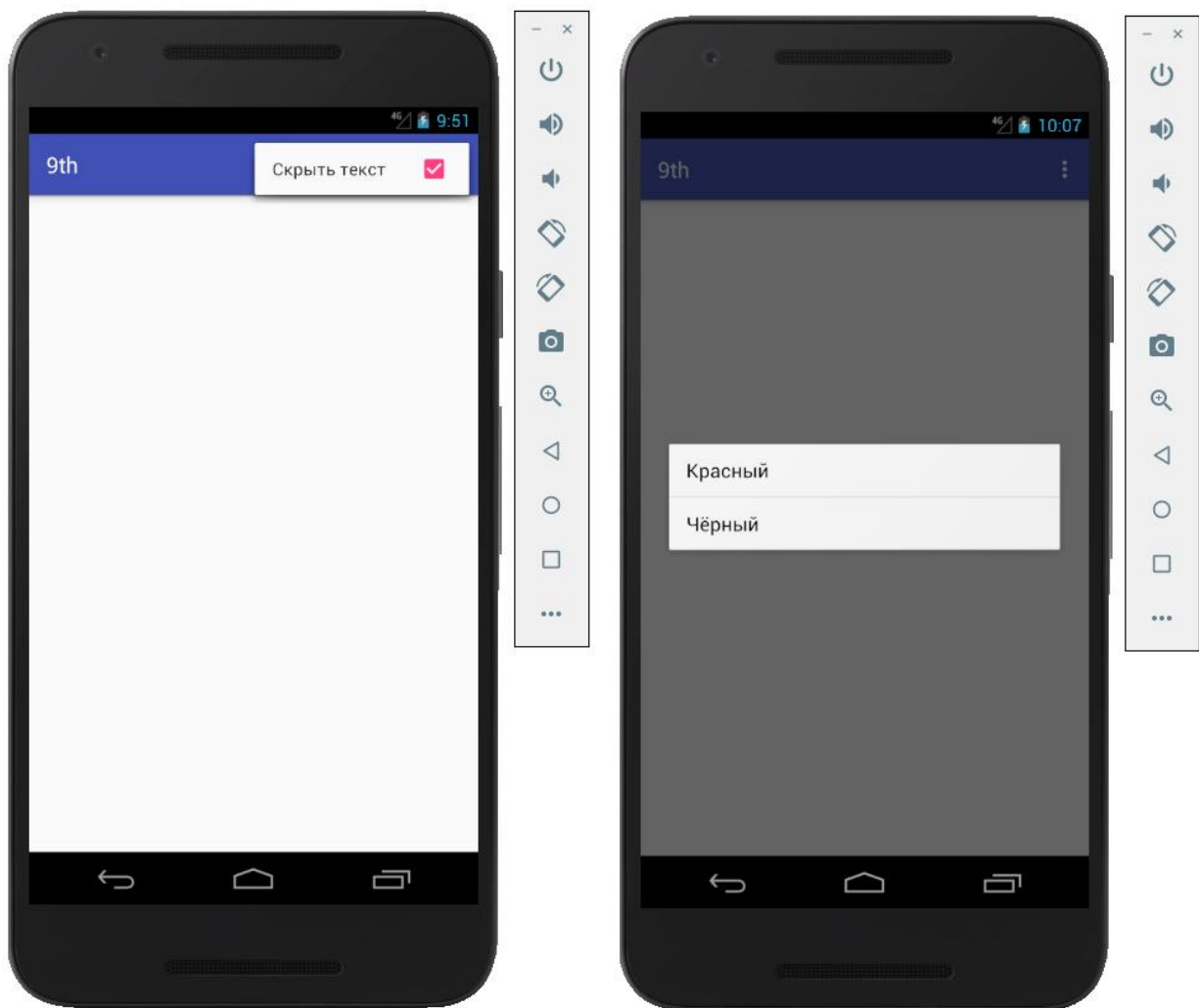


Рисунок 39 – Опциональное меню Рисунок 40 – Контекстное меню

Вы наверняка заметили, что в этом задании не были упомянуты ФИО. Сделайте ещё один пункт опционального меню О программе, в котором выведите с помощью Toast следующее: Автор – ФИО, гр. (создайте новый строковый ресурс, как обычно), добавив в соответствующее место кода следующее:

```
Toast.makeText(MainActivity.this, R.string.fio, Toast.LENGTH_LONG).show();
```

В результате появится информация о разработчике программы и новый пункт меню, как показано на рисунках 41 и 42.

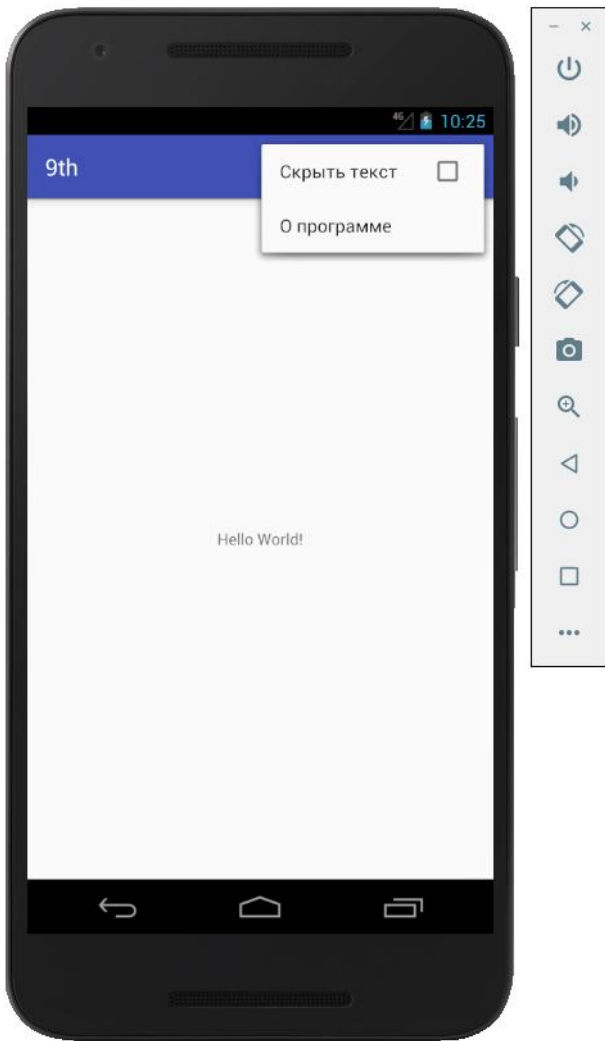


Рисунок 41 – Пункт "О программе"

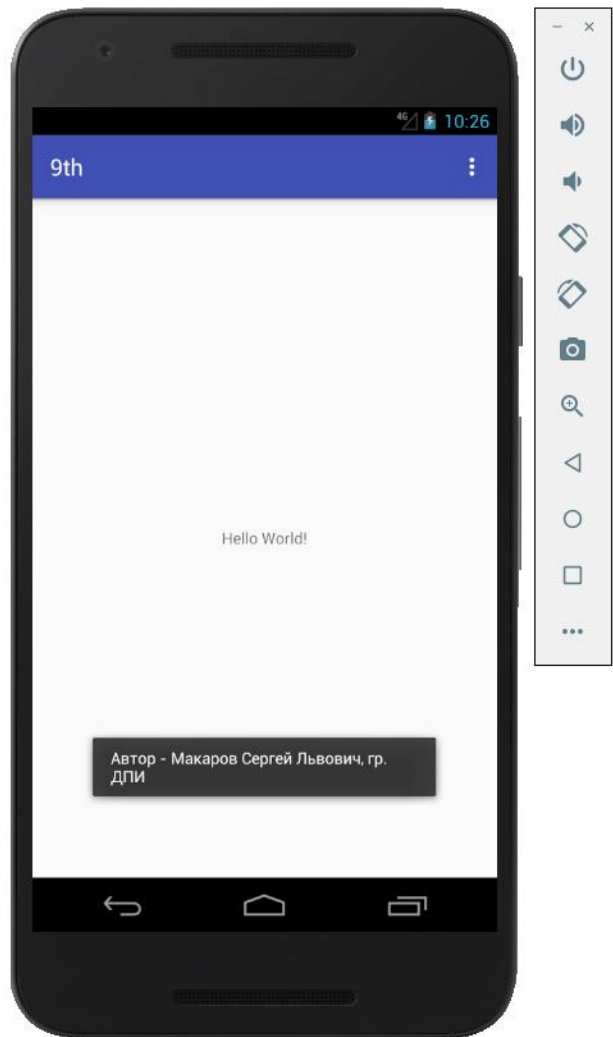


Рисунок 42 – Toast об авторе