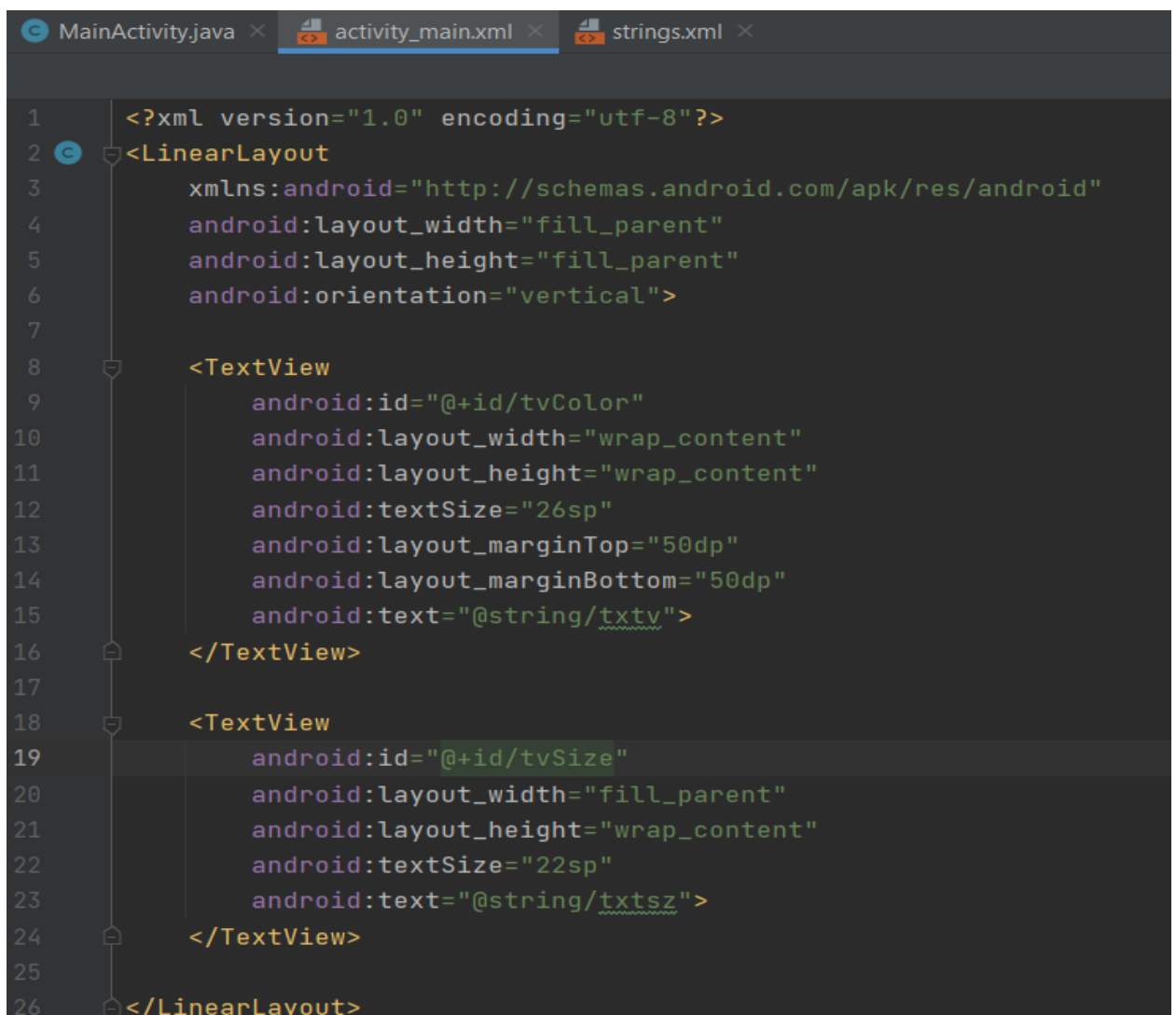


# Отчёт по проделанной работе на семинаре по мобильной разработке (11.10.2021).

(выполнил Валяев Георгий Анатольевич)

Для начала разработки простого примера работы контекстного меню поймём его функционал и определение: контекстное меню вызывается в Android длительным нажатием на каком-либо экранном компоненте. Обычно оно используется в списках, когда на экран выводится список однородных объектов и, чтобы выполнить действие с одним из этих объектов, мы вызываем контекстное меню для него.

Теперь создадим в activity\_main.xml два текстовых окна с описанием цвета и размера текста в LinearLayout:



```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      android:layout_width="fill_parent"
5      android:layout_height="fill_parent"
6      android:orientation="vertical">
7
8      <TextView
9          android:id="@+id/tvColor"
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         android:textSize="26sp"
13         android:layout_marginTop="50dp"
14         android:layout_marginBottom="50dp"
15         android:text="@string/txtv">
16     </TextView>
17
18     <TextView
19         android:id="@+id/tvSize"
20         android:layout_width="fill_parent"
21         android:layout_height="wrap_content"
22         android:textSize="22sp"
23         android:text="@string/txtsz">
24     </TextView>
25
26 </LinearLayout>
```

```
MainActivity.java × activity_main.xml × strings.xml ×
Edit translations for all locales in the translations editor.
1 <resources>
2
3     <string name="app_name">Context_menu</string>
4     <string name="txtv">Text color</string>
5     <string name="txtsz">Text size</string>
6
7 </resources>
```

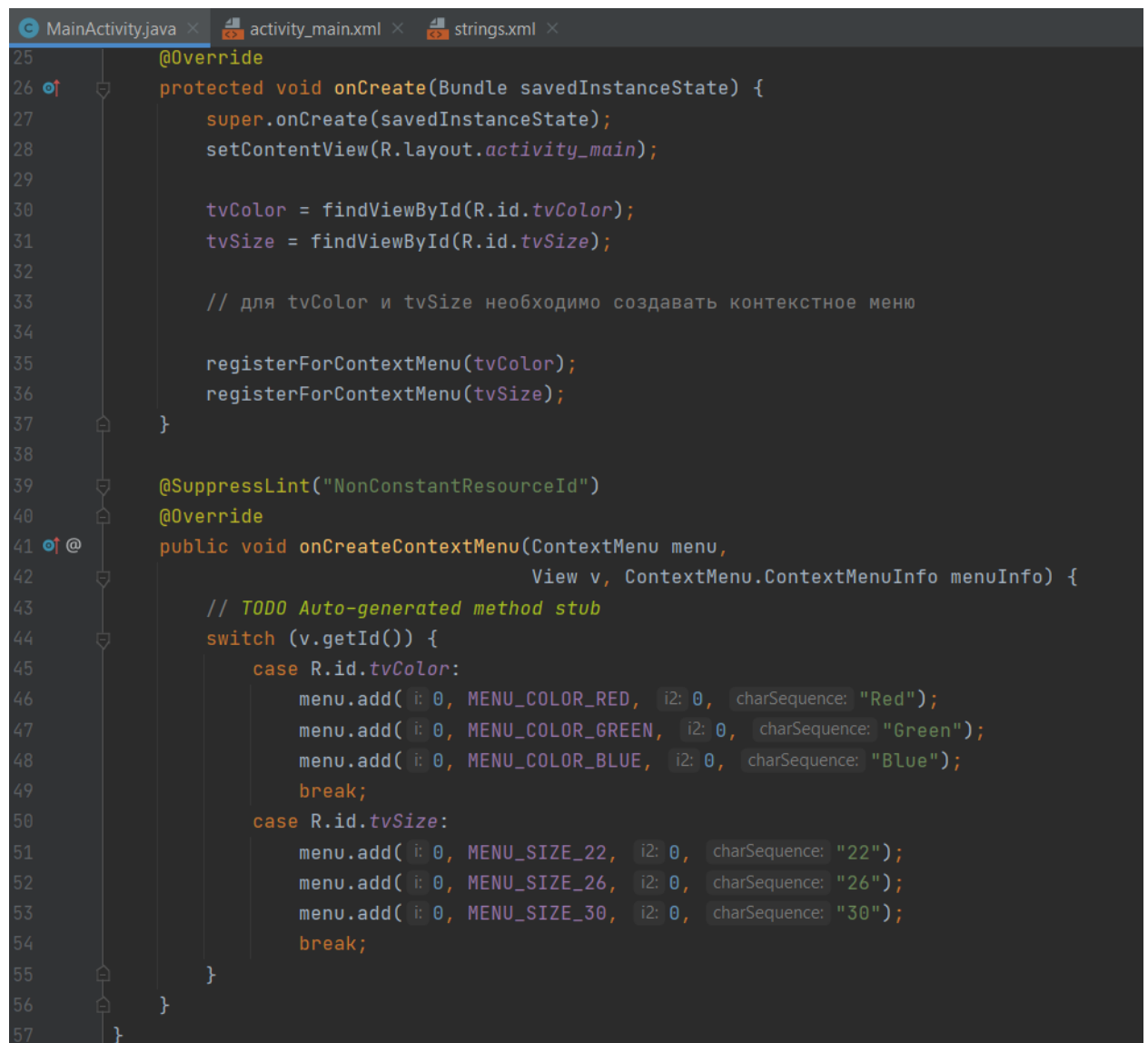
Далее начинаем кодить в MainActivity.java: добавляем 2 переменные типа TextView, глобальные переменные цветов и размеров текста, которые будут всплывать в качестве контекстного меню на экране приложения при долгом нажатии на сам элемент TextView.

И в главном методе класса получаем данные с TextView и создаём изначально для контекстное меню при помощи метода registerForContextMenu():

```
MainActivity.java × activity_main.xml × strings.xml ×
1 package com.example.context_menu;
2
3 import ...
4
10
11 public class MainActivity extends AppCompatActivity {
12
13     TextView tvColor, tvSize;
14
15     final int MENU_COLOR_RED = 1;
16     final int MENU_COLOR_GREEN = 2;
17     final int MENU_COLOR_BLUE = 3;
18
19     final int MENU_SIZE_22 = 4;
20     final int MENU_SIZE_26 = 5;
21     final int MENU_SIZE_30 = 6;
22
23
24     /** Called when the activity is first created. */
25     @Override
26     protected void onCreate(Bundle savedInstanceState) {
27         super.onCreate(savedInstanceState);
28         setContentView(R.layout.activity_main);
29
30         tvColor = findViewById(R.id.tvColor);
31         tvSize = findViewById(R.id.tvSize);
32
33         // для tvColor и tvSize необходимо создавать контекстное меню
34
35         registerForContextMenu(tvColor);
36         registerForContextMenu(tvSize);
37     }
```

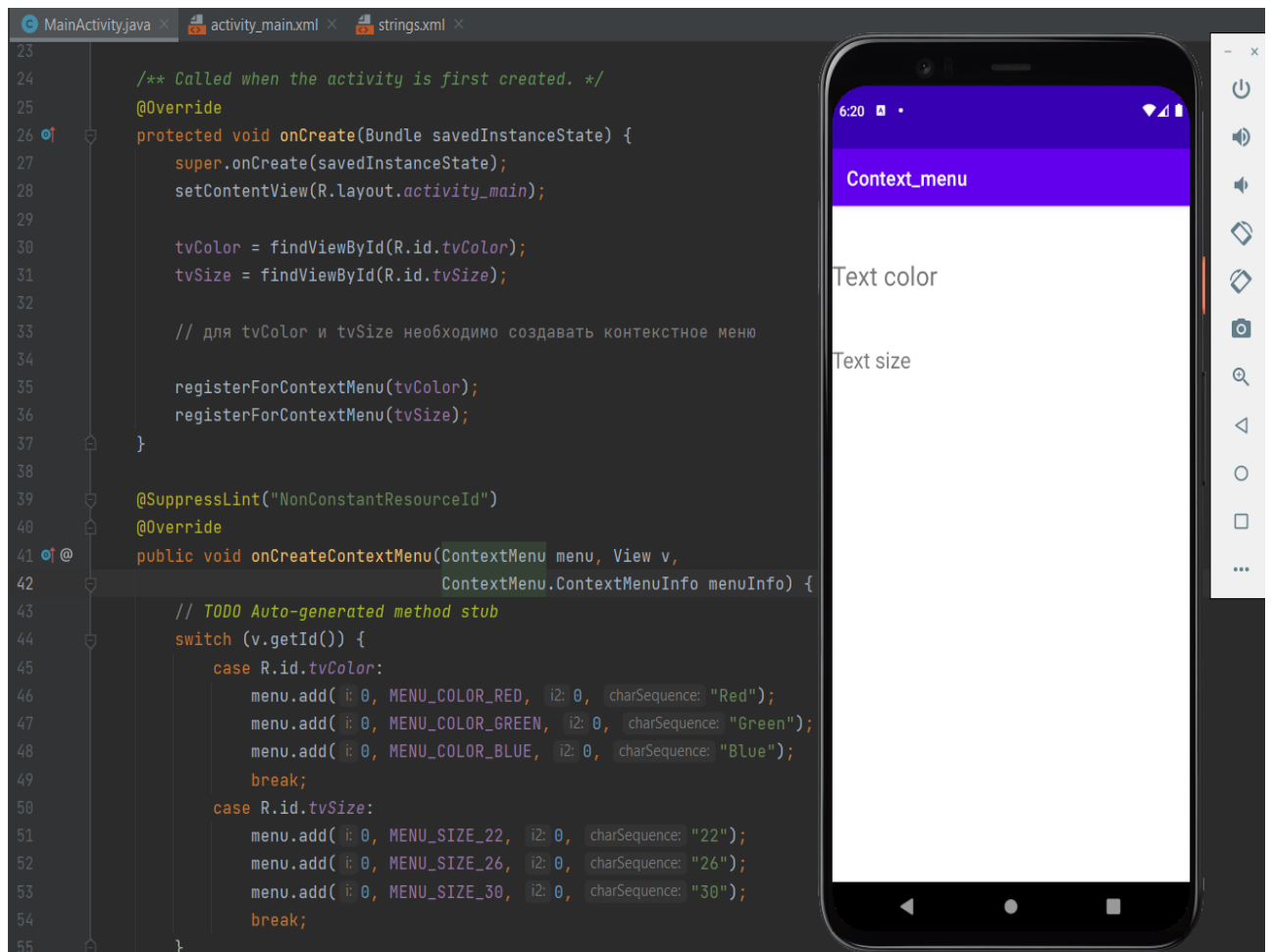
И затем создаём сам метод создания контекстного меню, который чем-то да похож на создание обыкновенного меню приложения, однако тут уже на входе нам понадобится само контекстное меню (именно в него мы и будем добавлять пункты при его появлении), элемент экрана, который был долго зажат пользователем и дополнительная информация при вызове контекстного меню какого-то конкретного элемента view.

В методе мы при помощи ID нажатого элемента находим его в данных и к контекстному меню добавляем пункты, которые появятся при вызове (цвета и размеры текста) без разделения на группы меню:

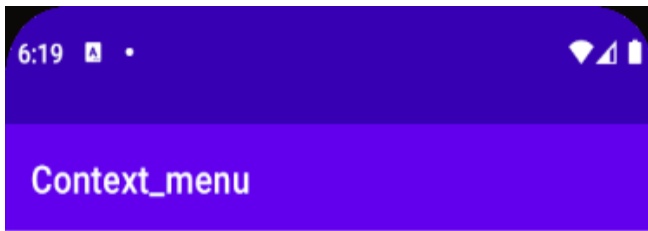


```
25      @Override
26      protected void onCreate(Bundle savedInstanceState) {
27          super.onCreate(savedInstanceState);
28          setContentView(R.layout.activity_main);
29
30          tvColor = findViewById(R.id.tvColor);
31          tvSize = findViewById(R.id.tvSize);
32
33          // для tvColor и tvSize необходимо создавать контекстное меню
34
35          registerForContextMenu(tvColor);
36          registerForContextMenu(tvSize);
37      }
38
39      @SuppressWarnings("NonConstantResourceId")
40      @Override
41      public void onCreateContextMenu(ContextMenu menu,
42                                     View v, ContextMenu.ContextMenuInfo menuInfo) {
43          // TODO Auto-generated method stub
44          switch (v.getId()) {
45              case R.id.tvColor:
46                  menu.add(0, MENU_COLOR_RED, 0, charSequence: "Red");
47                  menu.add(0, MENU_COLOR_GREEN, 0, charSequence: "Green");
48                  menu.add(0, MENU_COLOR_BLUE, 0, charSequence: "Blue");
49                  break;
50              case R.id.tvSize:
51                  menu.add(0, MENU_SIZE_22, 0, charSequence: "22");
52                  menu.add(0, MENU_SIZE_26, 0, charSequence: "26");
53                  menu.add(0, MENU_SIZE_30, 0, charSequence: "30");
54                  break;
55          }
56      }
57  }
```

Итоговый результат разработки меню:



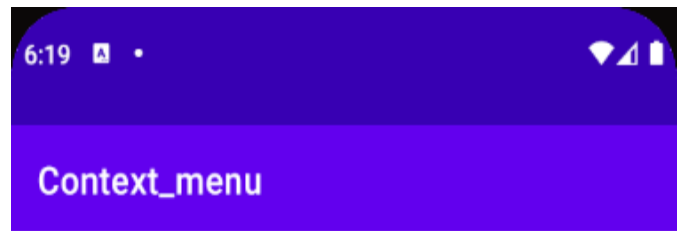
(дополнительно на след. странице)



Text color

Text size

Red  
Green  
Blue



Text color

Text size

22  
26  
30

