Министерство науки и высшего образования Российской Федерации

Севастопольский государственный университет

Кафедра ИС

Отчет

По дисциплине: «Разработка мобильных приложений»

Лабораторная работа №3

«Работа с ресурсами в ОС Android»

Выполнил

ст. гр. ИС/б-17-2о

Волобуев Ю.С.

Проверил:

Шишкевич В.Е.

Севастополь

2020

1 ЦЕЛЬ

Изучить организацию хранения различных типов **ресурсов**приложения Android и исследовать возможности их использования.

2 ВАРИАНТ ЗАДАНИЯ

Вариант 9:

|  |  |
| --- | --- |
| № варианта | Тип ресурса, который требуется использовать в проекте |
| 9 | Style;  Menu |

3 ХОД РАБОТЫ

1. Реализуем меню.

Создадим строковые ресурсы, содержащие названия пунктов меню в файле res/values/strings.xml:

<resources>  
 <string name="app\_name">TableLayoutApp</string>  
 <string name="LightStyle">Light Style</string>  
 <string name="DarkStyle">Dark Style</string>  
 <string name="GoldStyle">Gold Style</string>  
</resources>

Определим меню и все его пункты в ресурсе меню формата XML в файле res/menu/menu.xml:

<menu xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
  
 <item android:id="@+id/light\_style"  
 android:title="@string/LightStyle"  
 app:showAsAction="never"/>  
  
 <item android:id="@+id/dark\_style"  
 android:title="@string/DarkStyle"  
 app:showAsAction="never"/>  
  
 <item android:id="@+id/gold\_style"  
 android:title="@string/GoldStyle"  
 app:showAsAction="never"/>  
</menu>

Атрибут showAsAction определяет поведение меню, значение "never" обозначает, что элемент меню не должен выводиться в заголовке, а только во всплывающем меню.

Реализуем программно меню в MainActivity.java. Добавим метод onCreateOptionsMenu(), отвечающий за появление меню у активности:

@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
 MenuInflater inflater = getMenuInflater();  
 inflater.inflate(R.menu.*menu*, menu);  
 return super.onCreateOptionsMenu(menu);  
}

С помощью метода inflate(), в параметрах которого указан ресурс меню и объект класса Menu, записываем данные из XML-файла в объект inflater.

При запуске проекта в строке заголовка теперь появляется меню (см. рисунок 1):

Изображение выглядит как снимок экрана

Автоматически созданное описание Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 1 – Меню

Обработку нажатий пунктов меню реализуем в методе onOptionsItemSelected(), в котором для каждого пункта меню пропишем соответствующие действия:

@Override  
public boolean onOptionsItemSelected(@NonNull MenuItem item) {  
 Button[] arrayButton = new Button[]{press1, press2, press3, press4};  
 final TextView[] arrayTextView = new TextView[]{textView1, textView2, textView3, textView4, textView5};  
  
 switch(item.getItemId()){  
 case R.id.*light\_style*:{  
 setTheme(R.style.*LightTheme*);   
 tableLayout.setBackgroundColor(getResources().getColor(R.color.*LightThemeBack*));  
 for(int i = 0; i < arrayTextView.length; i++){  
 arrayTextView[i].setBackgroundColor(getResources().getColor(R.color.*LightThemeTextView*));  
 }

for(int i = 0;i < arrayButton.length; i++){  
 arrayButton[i].setBackgroundColor(getResources().getColor(R.color.*LightThemeButton*));

}  
 break;  
 }

case R.id.*dark\_style*:{  
 setTheme(R.style.*DarkTheme*); tableLayout.setBackgroundColor(getResources().getColor(R.color.*DarkThemeBack*));  
 for(int i = 0; i < arrayTextView.length; i++){  
 arrayTextView[i].setBackgroundColor(getResources().getColor(R.color.*DarkThemeTextView*));  
 }  
 for(int i = 0;i < arrayButton.length; i++){  
 arrayButton[i].setBackgroundColor(getResources().getColor(R.color.*DarkThemeButton*));  
 }  
   
 break;  
 }

case R.id.*gold\_style*:{

setTheme(R.style.*GoldTheme*);  
tableLayout.setBackgroundColor(getResources().getColor(R.color.*GoldThemeBack*));  
 for(int i = 0; i < arrayButton.length; i++){  
 arrayButton[i].setBackgroundColor(getResources().getColor(R.color.*GoldThemeButton*));  
 }  
 for(int i = 0; i < arrayTextView.length; i++){  
 arrayTextView[i].setBackgroundColor(getResources().getColor(R.color.*GoldThemeTextView*));

}  
 break;  
 }  
 }

TypedValue typedValue = new TypedValue();  
 Resources.Theme theme = getTheme();  
  
 theme.resolveAttribute(android.R.attr.*windowBackground*, typedValue, true);  
 getWindow().getDecorView().setBackgroundColor(typedValue.data);  
  
 theme.resolveAttribute(android.R.attr.*colorPrimary*, typedValue, true);  
 getSupportActionBar().setBackgroundDrawable(new ColorDrawable(typedValue.data));  
  
 theme.resolveAttribute(android.R.attr.*colorPrimaryDark*, typedValue, true);  
 Window window = getWindow();  
 window.addFlags(WindowManager.LayoutParams.*FLAG\_DRAWS\_SYSTEM\_BAR\_BACKGROUNDS*);  
 window.setStatusBarColor(typedValue.data);  
  
 return super.onOptionsItemSelected(item);  
 }  
}

Каждый пункт меню устанавливает новую тему и стили для компонентов приложения. Все стили и темы описаны в файле ресурсов res/values/styles.xml, в который вдобавок поместили все стили первоначальной страницы, повторяющиеся для одинаковых элементов:

<resources>  
 <!-- Base application theme. -->  
 <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">  
 <!-- Customize your theme here. -->  
 <item name="colorPrimary">@color/colorPrimary</item>  
 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>  
 <item name="colorAccent">@color/colorAccent</item>  
 </style>  
  
 <style name="imageStyle">  
 <item name="android:layout\_width">177dp</item>  
 <item name="android:layout\_height">146dp</item>  
 <item name="android:layout\_marginTop">50dp</item>  
 <item name="android:layout\_marginBottom">20dp</item>  
 </style>  
  
 <style name="textViewStylePrimary">  
 <item name="android:background">@color/textViewBack</item>  
 <item name="android:textAlignment">center</item>  
 <item name="android:textColor">@color/Black</item>  
 <item name="android:textSize">22sp</item>  
 </style>  
  
 <style name="ButtonPrimary">  
 <item name="android:layout\_marginBottom">10dp</item>  
 <item name="android:background">@color/buttonPrimaryBack</item>  
 <item name="android:textSize">24sp</item>  
 </style>  
  
 <style name="LightTheme" parent="@android:style/Theme.Material.Light">  
 <item name="android:colorPrimaryDark">@color/LightThemePrimaryDark</item>  
 <item name="android:colorPrimary">@color/LightThemePrimary</item>  
 </style>  
  
 <style name="DarkTheme" parent="@android:style/Theme.Material">  
  
 </style>  
  
 <style name="GoldTheme" parent="@android:style/Theme.Material.Light">  
 <item name="android:colorPrimary">@color/GoldThemePrimary</item>  
 <item name="android:colorPrimaryDark">@color/GoldThemePrimaryDark </item>  
 </style>  
</resources>

Все цвета описаны в файле res/values/colors.xml:

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="colorPrimary">#008577</color>  
 <color name="colorPrimaryDark">#00574B</color>  
 <color name="colorAccent">#D81B60</color>  
 <color name="Black">#000000</color>  
 <color name="textViewBack">#c8e6c9</color>  
 <color name="buttonPrimaryBack">#81c784</color>  
  
 <color name="DarkThemeBack">#302F2F</color>  
 <color name="DarkThemeButton">#493C3C</color>  
 <color name="DarkThemeTextView">#998D8D</color>  
  
 <color name="LightThemeBack">#e1f5fe</color>  
 <color name="LightThemeButton">#4fc3f7</color>  
 <color name="LightThemeTextView">#b3e5fc</color>  
 <color name="LightThemePrimary">#91C3DB</color>  
 <color name="LightThemePrimaryDark">#82b3c9</color>  
  
 <color name="GoldThemePrimary">#f9bf2d</color>  
 <color name="GoldThemePrimaryDark">#c28f00</color>  
 <color name="GoldThemeButton">#D58F23</color>  
 <color name="GoldThemeTextView">#F3C77A</color>  
 <color name="GoldThemeBack">#F0E59C</color>  
   
</resources>

1. Протестируем работу приложения. Установим светлую тему, то есть первый пункт меню (см. рисунок 2):

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 2 – Скриншот приложения

Установим тёмную и золотую тему (см. рисунок 3)

Изображение выглядит как монитор, сидит, улица, экран

Автоматически созданное описание Изображение выглядит как снимок экрана, рисунок

Автоматически созданное описание

Рисунок 3 – Скриншоты приложения

ВЫВОДЫ

В ходе данной лабораторной работы была изучена организация хранения различных типов **ресурсов**приложения Android, а также исследованы возможности их использования. Были изучены особенности работы с использованием стилей и тем, а также создание всплывающего меню.