

Молдавский Государственный Университет
Факультет Математики и Информатики
Департамент Информатики

Лабораторная работа Nr.3
Тема: “Графический интерфейс.
Фрагменты/Виджеты.”

Проверил; Георгий Латул

Выполнил: Константин Рунтов

Группа: i 1802

Кишинев 2021

Содержание

Содержание	2
Условие задачи	3
Выполнение задачи	3
1. Создание приложения	3
1.1 Создание скелета приложения	3
1.2 Создание элемента Widget	4
1.2 Создание конфигурационного окна виджета	7
2. Работа приложения	10
Вывод	12

Условие задачи

Создание Android приложения, которое будет иметь виджет. Данный виджет должен иметь какой-нибудь функционал.

Выполнение задачи

1. Создание приложения

1.1 Создание скелета приложения

Первым делом созданием скелет нашего приложения на основе макета с пустой активностью. На основе данного макета будем выстраивать вид и логику нашего приложения.

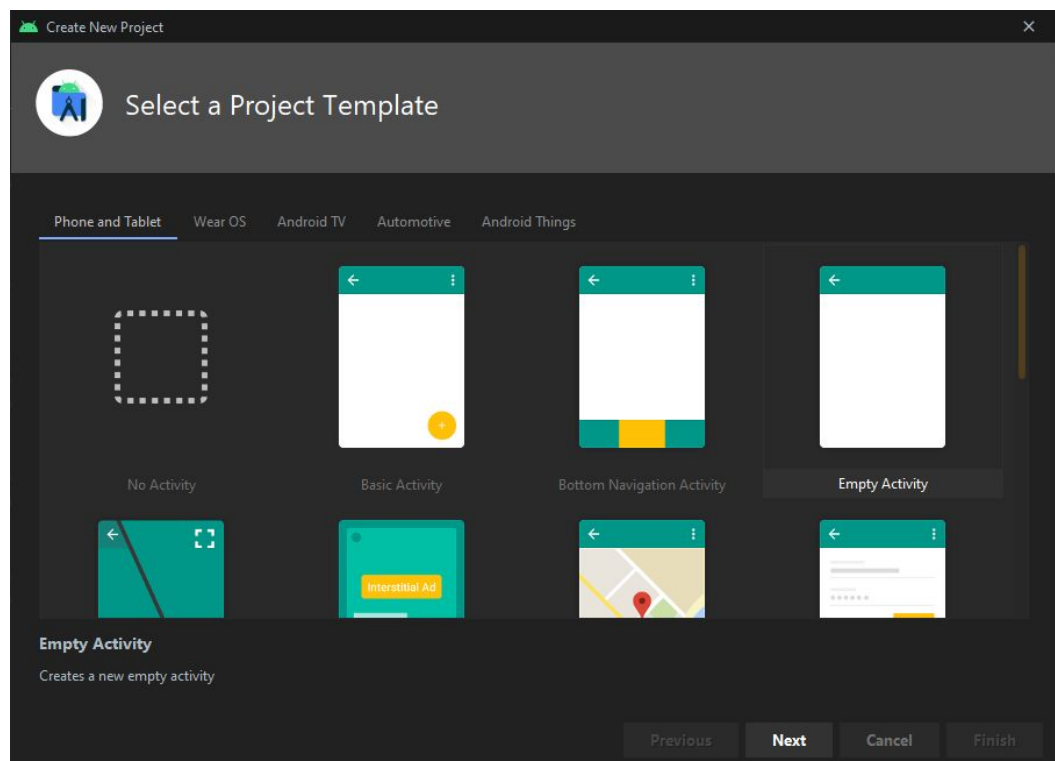


Рис. 1.1 Окно выбора шаблона приложения

1.2 Создание элемента Widget

Создание виджета может быть выполнено посредством генерации необходимых элементов, так называемый “*starter pack*”, или же путём ручного создания необходимых файлов и внесения изменений в определенные файлы. Для выполнения данной работы я выбрал второй вариант. Требуется создать следующие файлы:

- *activity_widget.xml* в папке layout
- *activity_widget_config.xml* в папке layout
- *xml* папку
- *widget_info.xml* в папке xml

Первым делом создадим файл описывающий облик нашего виджета и настроим его так, как мы этого желаем.

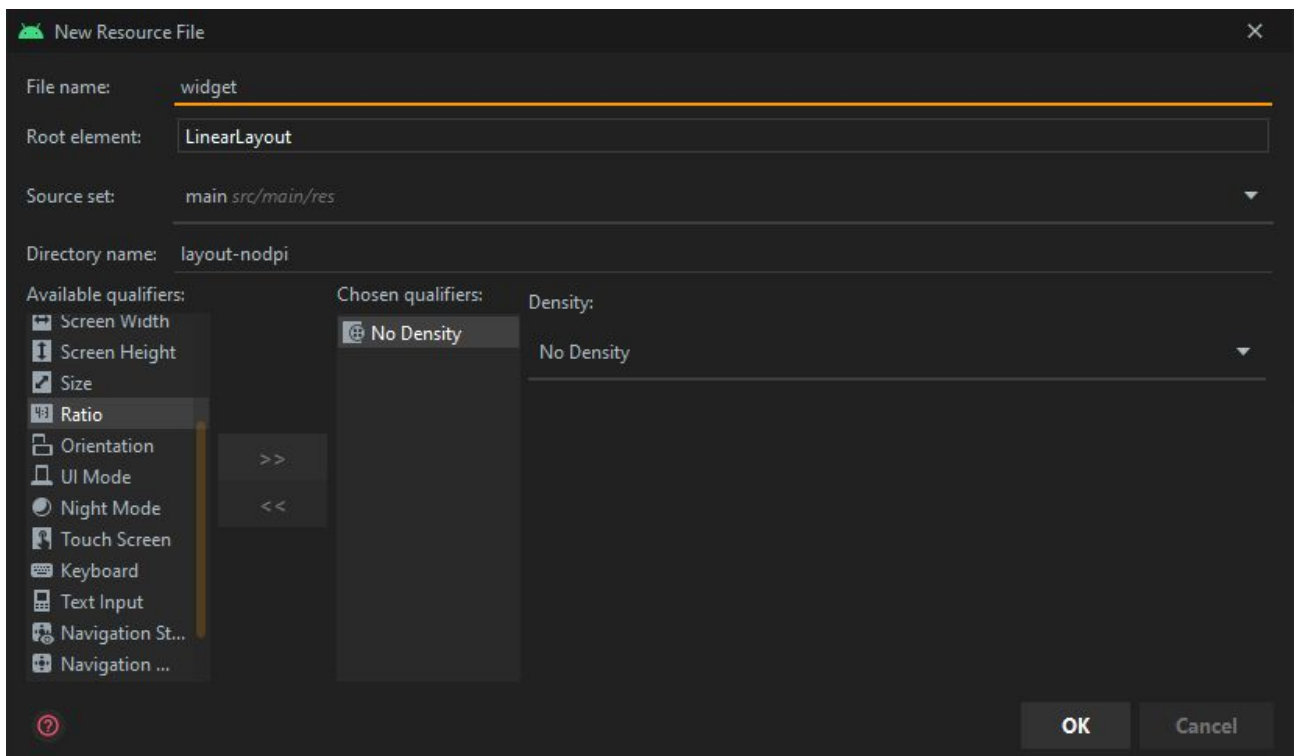


Рис. 1.2 Окно создания xml ресурс файла отвечающего за облик самого виджета

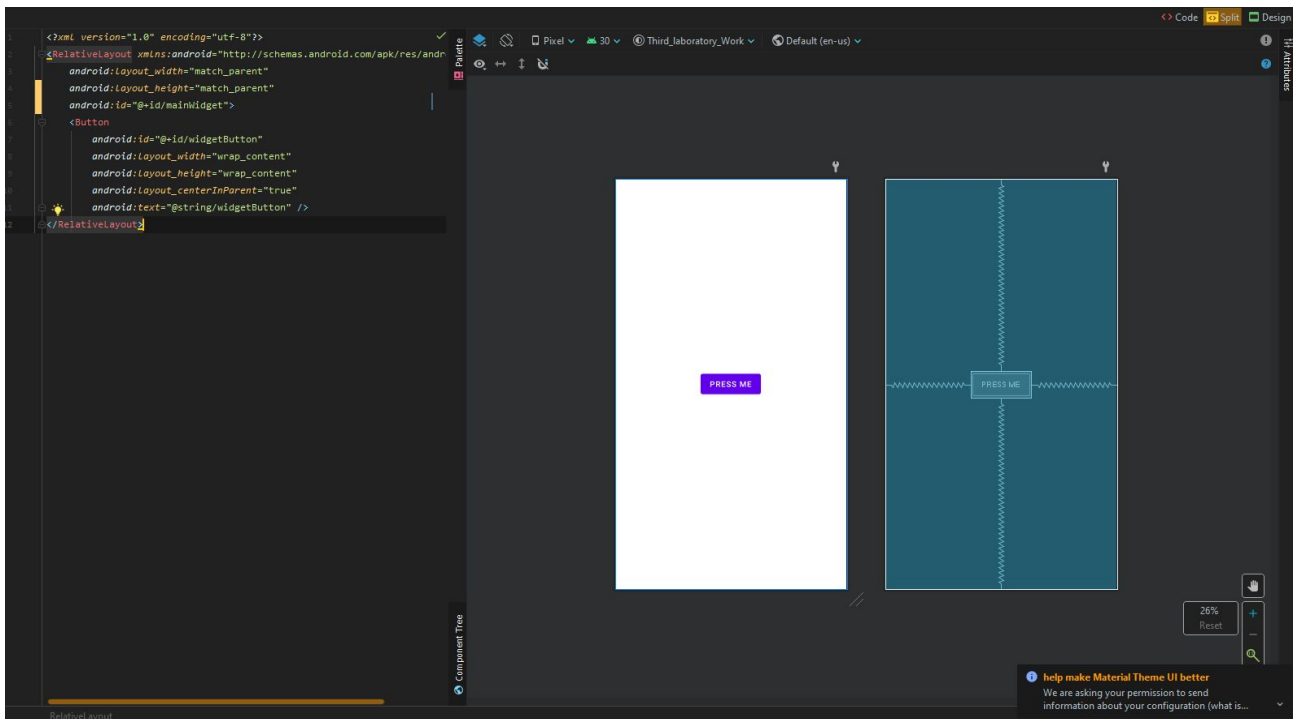


Рис. 1.3 Конструктор отображения нашего виджета

После создания скелета нашего виджета, приступим к написанию внутренностей, которые будут управлять им, а точнее java класса.

```
package cutemonster.third_laboratory_work;

import android.app.PendingIntent;
import android.appwidget.AppWidgetManager;
import android.appwidget.AppWidgetProvider;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.widget.RemoteViews;

import static
cutemonster.third_laboratory_work.Widget_Config.KEY_BUTTON_TEXT;
import static cutemonster.third_laboratory_work.Widget_Config.SHARED_PREFS;

public class WidgetProvider extends AppWidgetProvider {
    @Override
```

```

    public void onUpdate(Context context, AppWidgetManager appWidgetManager,
int[] appWidgetIds) {
        for (int appWidgetId : appWidgetIds) {
            /// Creating intent which will be called when listener is
triggered

            Intent intent = new Intent(context, MainActivity.class);
            PendingIntent pendingIntent = PendingIntent.getActivity(context,
0, intent, 0);

            /// Creating to save preferences between invokes
            SharedPreferences prefs =
context.getSharedPreferences(SHARED_PREFS, Context.MODE_PRIVATE);
            String buttonText = prefs.getString(KEY_BUTTON_TEXT + appWidgetId,
"Press me");

            /*
                Creating RemoteViews object by which we can access and make
changes on activities
                and their items. We can't do this like usual? cause here we
access them through stream
            */
            RemoteViews views = new RemoteViews(context.getPackageName(),
R.layout.activity_widget);
            views.setOnClickPendingIntent(R.id.widgetButton, pendingIntent);
            views.setCharSequence(R.id.widgetButton, "setText", buttonText);

            /// Calling updateAppWidget method from AppWidgetManager to update
specified widget
            // by passed view settings
            appWidgetManager.updateAppWidget(appWidgetId, views);
        }
    }
}

```

Данный класс содержит лишь переопределенный метод *onUpdate*, который вызывается каждый раз, как происходит изменение нашего виджета или же при задании определенного таймаута. Цикл *for* используется для того, что пройти по массиву созданных виджетов, так как допускается создать множество копий виджета на экране.

Далее создаем файл *widget_info.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<appwidget-provider
xmlns:android="http://schemas.android.com/apk/res/android"
    android:configure="cutemonster.third_laboratory_work.Widget_Config"
    android:initialLayout="@layout/activity_widget"
    android:minHeight="40dp"
    android:minResizeWidth="40dp"
    android:minWidth="110dp"
    android:previewImage="@drawable/build"
    android:resizeMode="vertical|horizontal"
    android:updatePeriodMillis="3600000"
    android:widgetCategory="home_screen">

</appwidget-provider>
```

В данном файле указываем информацию о нашем виджете:

- Изображение используемое при предварительном просмотре виджета в списке всех виджетов доступных в системе
- Конфигурационный класс, который будет вызван перед созданием виджета
- Параметры размера
- Категорию

1.2 Создание конфигурационного окна виджета

Создадим файл представления окна конфигурации виджета и последующий класс, отвечающий за его работу. Данная активность будет иметь всего два элемента, так как у нас виджет простой и не требует большей настройки.

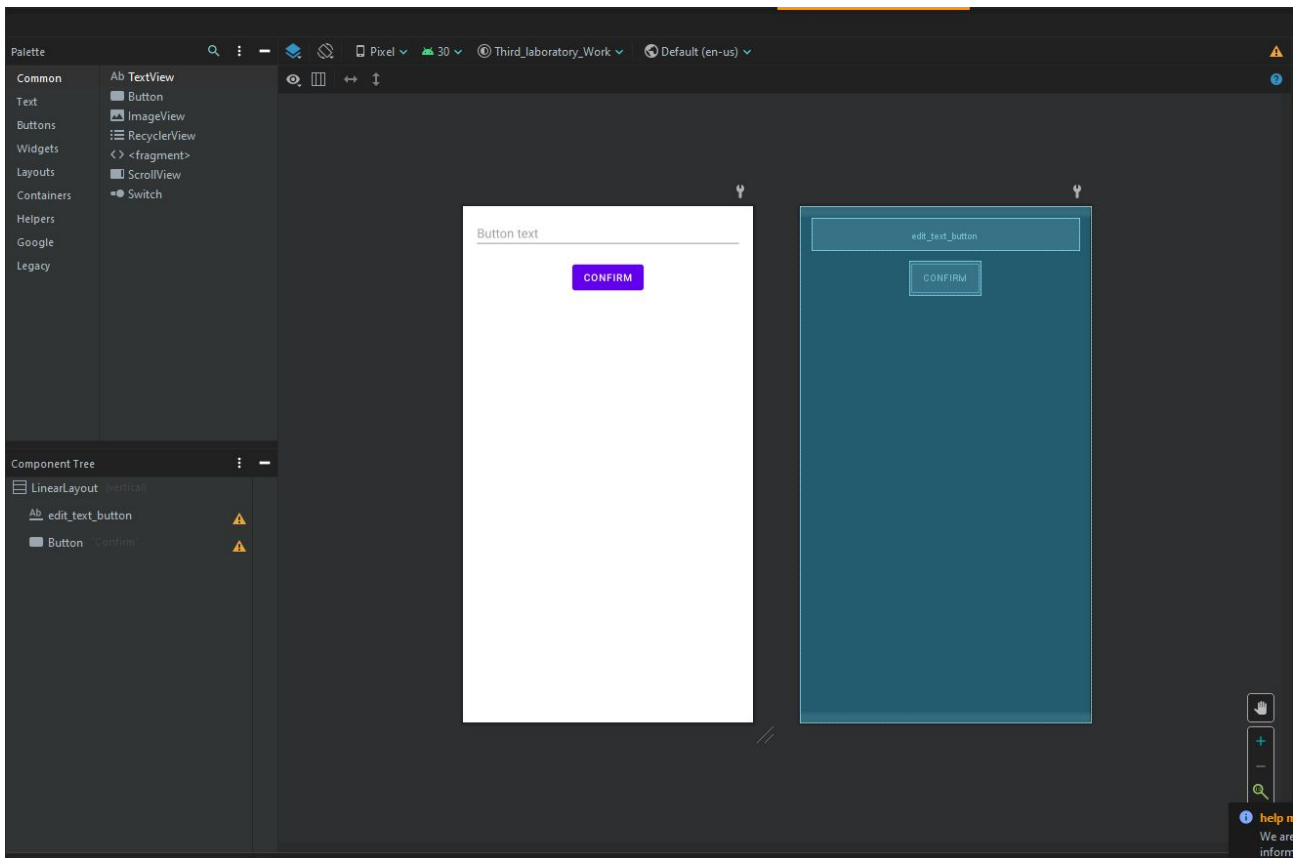


Рис. 1.4 Окно конфигурации виджета перед его созданием

Далее создаем класс отвечающий за данную активность в котором дополняем метод `onCreate` следующим блоком кода:

```
/// Getting extras from current intent
Bundle extras = getIntent().getExtras();
if (extras != null) {
    /// Getting widget_id from extra
    this.appWidgetId = extras.getInt(AppWidgetManager.EXTRA_APPWIDGET_ID,
        AppWidgetManager.INVALID_APPWIDGET_ID);
}
Intent resultValue = new Intent();
///Filling intent with current widget id
resultValue.putExtra(AppWidgetManager.EXTRA_APPWIDGET_ID, this.appWidgetId);
setResult(RESULT_CANCELED, resultValue);

if (appWidgetId == AppWidgetManager.INVALID_APPWIDGET_ID) {
    /// Finish widget configuration activity if widget id belongs to invalid
```



```

widget
    finish();
}
/// Getting view button instance
this.editTextButton = findViewById(R.id.edit_text_button);

```

Также определяем метод *confirmConfiguration* в котором записываем следующий блок кода:

```

/// Get instance of WindowManager to make changes on widget
AppWidgetManager appWidgetManager = AppWidgetManager.getInstance(this);

// Setting intent which will be called by widget
Intent intent = new Intent(this, MainActivity.class);
Intent resultValue = new Intent();
PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent, 0);
String buttonText = this.editTextButton.getText().toString();
RemoteViews views = new RemoteViews(this.getPackageName(),
R.layout.activity_widget);

/// Creating of shared preferences and editor to save configuration between
calling new widget
SharedPreferences prefs = getSharedPreferences(SHARED_PREFS, MODE_PRIVATE);
SharedPreferences.Editor editor = prefs.edit();

/// Configuring widget view and updating widget
views.setOnClickPendingIntent(R.id.widgetButton, pendingIntent);
views.setCharSequence(R.id.widgetButton, "setText", buttonText);
appWidgetManager.updateAppWidget(appWidgetId, views);
editor.putString(KEY_BUTTON_TEXT + appWidgetId, buttonText);
editor.apply();

/// Saving result with RESULT_OK code and finishing widget config activity
resultValue.putExtra(AppWidgetManager.EXTRA_APPWIDGET_ID, appWidgetId);
setResult(RESULT_OK, resultValue);
finish();

```

2. Работа приложения

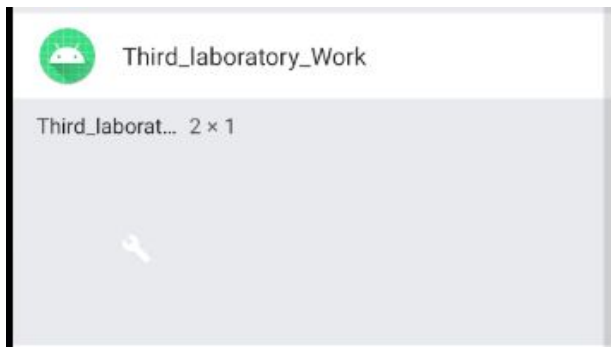


Рис. 1.4 Отображение нашего виджета в общем списке

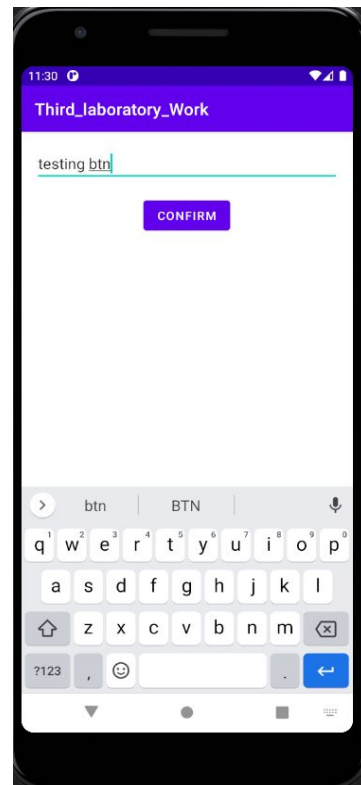


Рис. 1.5 Вызванная активность конфигурации при создании виджета

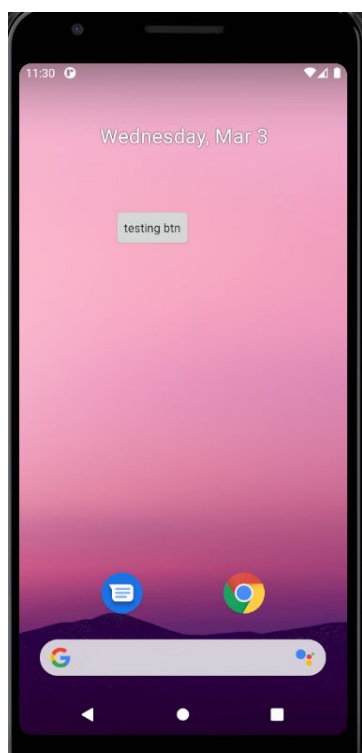


Рис. 1.6 Сконфигурированный виджет с отображением нашего текста



Рис. 1.7 Вызванная активность при нажатии на кнопку виджета

Вывод

Выполнение данной лабораторной работы дало понятие того, как создаются виджеты для Android приложения, а также как работать с ними.

Было создано приложение с тремя активностями:

1. Базовая активность приложения
2. Конфигурационная активность для виджета
3. Сам виджет