Пермский филиал федерального государственного автономного образовательного учреждения высшего образования   
«Национальный исследовательский университет   
«Высшая школа экономики»

*Факультет экономики, менеджмента и бизнес-информатики*

Чепоков Елизар Сергеевич

**ПРОЕКТ 0**

*Отчет*

студента образовательной программы «Программная инженерия»

по направлению подготовки *09.03.04 Программная инженерия*

Руководитель:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

А.В. Яборов

Пермь, 2021 год

**Оглавление**

[**Постановка задачи** 3](#_Toc63722438)

[**Первая часть работы** 4](#_Toc63722439)

[**Вторая часть работы** 7](#_Toc63722440)

# **Постановка задачи**

1. Первая часть работы (demo):
   1. Ограничить ввод в поле для ввода только числовыми значениями.
   2. Добавить логику для нажатия на кнопку Button2. Считать произведение всех четных чисел и вывести результат на экран по аналогии с кнопкой Button1.
   3. Добавить валидацию на диапазон вводимого значения, диапазон придумать самостоятельно. В случае выхода за диапазон выводить Toast с текстом о выходе за диапазон.
2. Вторая часть работы (base - пример одного экрана приложения):
   1. Файл с логотипом скачать по ссылке https://www.hse.ru/data/2012/01/19/1263884289/logo\_%D1%81\_hse\_cmyk\_e.png
   2. Поместить скачанный файл в папку drawable-xxxhdpi
   3. Сделать верстку экрана как на картинке справа. Создать новую activity и файл с версткой для нее activity\_main.xml. Запускать по умолчанию данную activity
   4. Основные цвета приложения указать в файле стилей themes.xml

colorPrimary #003399

colorPrimaryVariant #011F5A

colorSecondary #4080ff

colorSecondaryVariant #0C5DFD

* 1. Тексты для кнопок вынести в файл strings.xml
  2. Добавить показ Toast при нажатии на кнопки с информацией о нажатой кнопке

# **Первая часть работы**

Задача: Разработать программу «Demo» основываясь на презентации

1. Ограничить ввод в поле для ввода только числовыми значениями.

Сначала добавляем элементы интерфейса в файл activity\_demo.xml

<LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="15dp"  
 android:gravity="center\_horizontal"  
 android:orientation="vertical"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <EditText  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:id="@+id/number"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"/>  
  
 <Button  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/button1" />  
  
 <Button  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:id="@+id/button2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"

android:text="@string/button2" />  
  
 <TextView  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:id="@+id/result"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:textSize="24sp"  
 tools:text="@string/result\_string" />  
</LinearLayout>

Затем добавляем надстройку для элемента «EditText» (выделено желтым)

<EditText  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:id="@+id/number"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="number"  
 android:hint="@string/editText"  
 android:autofillHints="" />

1. Добавить логику для нажатия на кнопку Button2. Считать произведение всех четных чисел и вывести результат на экран по аналогии с кнопкой Button1.

Добавляем логику по нажатию кнопки «Button2» в файле DemoActivity.java

@Override protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_demo*);

...

button2.setOnClickListener(new View.OnClickListener() {  
 @Override public void onClick(View v) { clickButton2(); }  
});

private void clickButton2() {  
 //Log.d("Status","Button 2 pressed");  
 //Toast.makeText(DemoActivity.this,"Button 2 pressed",Toast.LENGTH\_SHORT).show();  
 String numberVal = number.getText().toString();  
 //number.getText().clear();  
 if (numberVal.isEmpty()) {  
 number.setText("0");  
 numberVal = "0";  
 }  
 int count = Integer.*parseInt*(numberVal);  
  
 List<Integer> list = new ArrayList<>();  
 for (int i = 0; i < count; i++) {  
 if (i % 2 == 0) list.add(i);  
 else continue;  
 }  
  
 int sum = 0;  
 if (android.os.Build.VERSION.*SDK\_INT* >= android.os.Build.VERSION\_CODES.*N*) {  
 sum = list.stream().mapToInt(Integer::intValue).sum();  
 }  
 result.setText(String.*format*("Result: %d", sum));  
}

Добавляем логику по нажатию кнопки «Button1» в файле DemoActivity.java

@Override protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_demo*);

...

button1.setOnClickListener(new View.OnClickListener() {  
 @Override public void onClick(View v) { clickButton1(); }  
});

private void clickButton1() {  
 String numberVal = number.getText().toString();  
 //number.getText().clear();  
 if (numberVal.isEmpty()) {  
 number.setText("0");  
 numberVal = "0";  
 }  
 int count = Integer.*parseInt*(numberVal);  
  
 List<Integer> list = new ArrayList<>();  
 for (int i = 0; i<count;i++) list.add(i+1);  
  
 int sum = list.stream().mapToInt(Integer::intValue).sum();  
 result.setText(String.*format*("Result: %d", sum));  
}

1. Добавить валидацию на диапазон вводимого значения, диапазон придумать самостоятельно. В случае выхода за диапазон выводить Toast с текстом о выходе за диапазон.

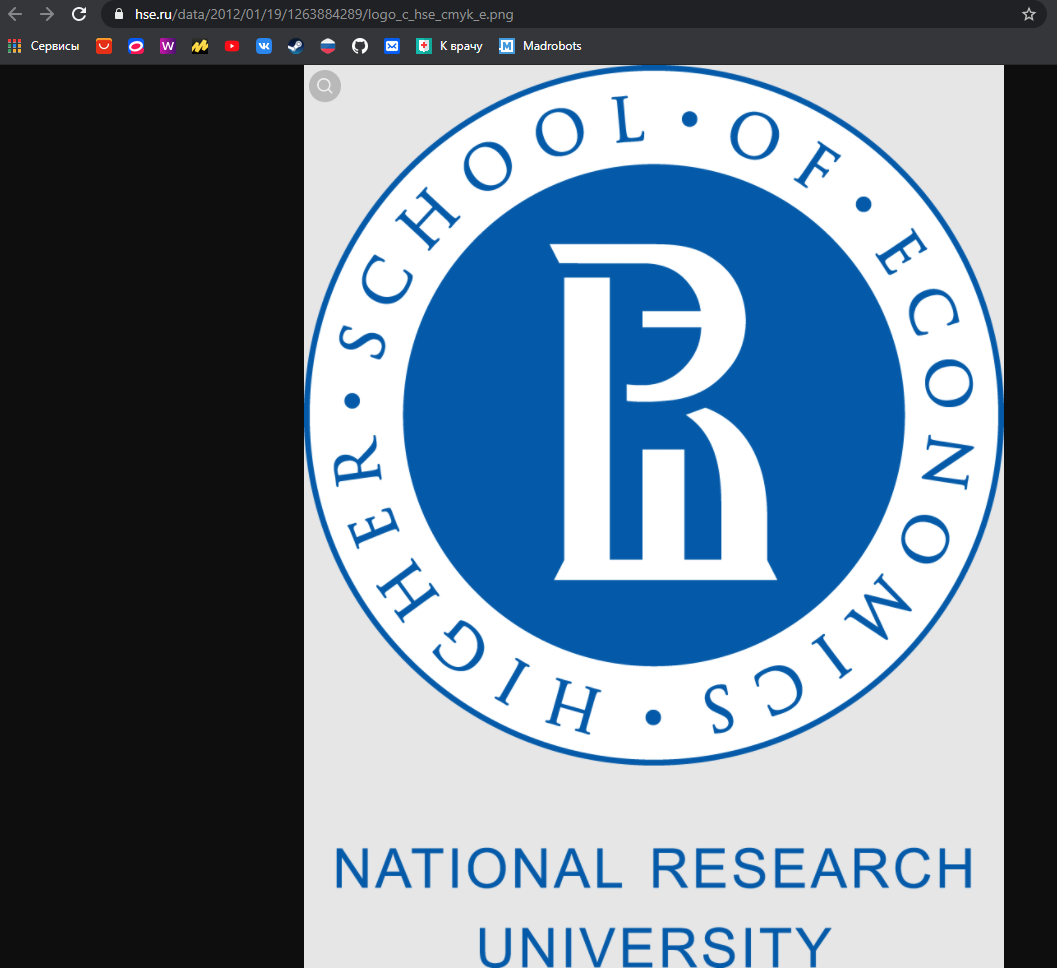
Изменяем логику при нажатии на кнопку с добавлением класса «Toast»

button1.setOnClickListener(new View.OnClickListener() {  
 @Override public void onClick(View v) { Clicker("b1"); }  
});  
button2.setOnClickListener(new View.OnClickListener() {  
 @Override public void onClick(View v) { Clicker("b2"); }  
});

private void Clicker(String s){  
 String check\_null = number.getText().toString();  
 Long longstr = Long.*parseLong*(check\_null);  
 if (check\_null!=""||check\_null!=null){  
 if (longstr > 100){  
 Toast.*makeText*(DemoActivity.this, "Введите от 0 до 100", Toast.*LENGTH\_SHORT*).show();  
 number.getText().clear();  
 }  
 else{  
 if (s == "b1"){ clickButton1(); }  
 else if(s == "b2"){ clickButton2(); }  
 else{ Log.*e*("Status","Error"); }  
 }  
 }  
 else{  
 Toast.*makeText*(DemoActivity.this, "Введите от 0 до 100", Toast.*LENGTH\_SHORT*).show();  
 number.getText().clear();  
 }  
}

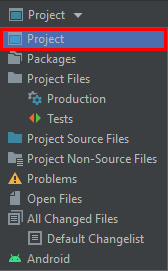
# **Вторая часть работы**

Задача: Разработать программу «Base» основываясь на презентации

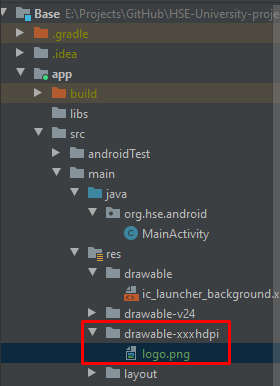
1. Файл с логотипом скачать по ссылке <https://www.hse.ru/data/2012/01/19/1263884289/logo_%D1%81_hse_cmyk_e.png>

***Рисунок 1 – Логотип***

1. Поместить скачанный файл в папку drawable-xxxhdpi

Переключаемся на перспективу отображения «Пакеты», так как при перспективе «Android» данные папки не видно.

***Рисунок 2 – Перспективы***

Так как папка «drawable-xxxhdpi» отсутствовала (была только «mipmap-xxxhdpi») пришлось ее создать

***Рисунок 3 – Строение проекта***

1. Сделать верстку экрана как на рисунке 4. Создать новую activity и файл с версткой для нее activity\_main.xml. Запускать по умолчанию данную activity

***Рисунок 5 – Верстка***

***Рисунок 4 – Пример***

1. Основные цвета приложения указать в файле стилей themes.xml

colorPrimary #003399

colorPrimaryVariant #011F5A

colorSecondary #4080ff

colorSecondaryVariant #0C5DFD

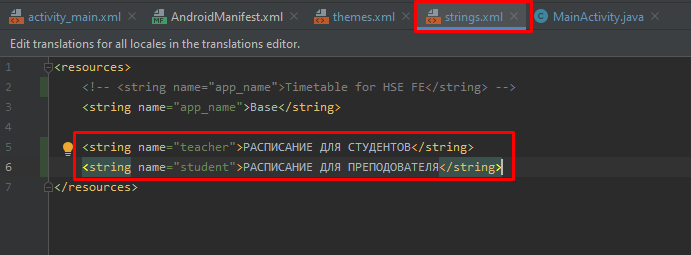
Добавляем новую палитру цветов в файле themes.xml:

<style name="Theme.HSE" parent="Theme.MaterialComponents.DayNight.DarkActionBar">  
 <!-- Primary brand color. -->  
 <item name="colorPrimary">#003399</item>  
 <item name="colorPrimaryVariant">#011F5A</item>  
 <item name="colorOnPrimary">@color/white</item>  
 <!-- Secondary brand color. -->  
 <item name="colorSecondary">#4080ff</item>  
 <item name="colorSecondaryVariant">#0C5DFD</item>  
 <item name="colorOnSecondary">@color/black</item>  
 <!-- Status bar color. -->  
 <item name="android:statusBarColor" tools:targetApi="l">?attr/colorPrimaryVariant</item>  
 <!-- Customize your theme here. -->  
</style>

В файле AndroidManifest.xml меняем палитру на нужную (выделено желтым)

<application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.HSE"> </application>

***Рисунок 6 – Итог***

1. Тексты для кнопок вынести в файл strings.xml

***Рисунок 7 – strings.xml***

1. Добавить показ Toast при нажатии на кнопки с информацией о нажатой кнопке

@Override  
protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 getSupportActionBar().show();  
  
 View button\_student = findViewById(R.id.*button\_student*);  
 View button\_teacher = findViewById(R.id.*button\_teacher*);  
  
 button\_student.setOnClickListener(new View.OnClickListener() {  
 @Override public void onClick(View v) { Toast.*makeText*(MainActivity.this, "РАСПИСАНИЕ ДЛЯ СТУДЕНТОВ", Toast.*LENGTH\_SHORT*).show(); }  
 });  
 button\_teacher.setOnClickListener(new View.OnClickListener() {  
 @Override public void onClick(View v) { Toast.*makeText*(MainActivity.this, "РАСПИСАНИЕ ДЛЯ ПРЕПОДОВАТЕЛЯ", Toast.*LENGTH\_SHORT*).show(); }  
 });  
}

***Рисунок 8 – 1 кнопка***

***Рисунок 9 – 2 кнопка***