**Министерство образования и науки Российской Федерации**

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа информационных технологий и робототехники

Отделение информационных технологий

Направление – Информационные системы и технологии

**Лабораторная работа №2**

по дисциплине: Программирование мобильных устройств и встраиваемых систем

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил: студент гр. 8И6А | \_\_\_\_\_\_\_\_\_\_\_ | Н. П. Шкулов | |
|  |  | |  |
| Проверил: ассистент ОИТ | \_\_\_\_\_\_\_\_\_\_\_ | | А. В. Погребной |

Томск 2020 г.

# Задание

Custom View содержит вертикальный список записей. Каждая запись состоит из трёх View. Первая – заголовок. Вторая – полоска, отображающая рейтинг (может быть в диапазоне от 0 до 10). Наивысший рейтинг отображается красным цветом, остальные чёрным. Последняя – текстовое представление рейтинга. Сверху над Custom View находится 2 EditText-а и кнопка. При нажатии на кнопку заголовок и оценка получаются из EditText-ов и добавляются в CustomView.

# Ход работы

Для выполнения работы нужно было создать кастомный View, который бы выводился в виде списка, состоящего из трех частей. Для этого был создан отдельный xml-файл лэйаута: сам он представляет собой линейный лэйаут с горизонтальным расположением элементов, внутри которого находится TextView, отображающий заголовок, ProgressBar для отображения рейтинга в виде полоски и ещё один TextView, представляющий значение рейтинга в числовом виде от 0 до 10.

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/item\_layout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:paddingTop="15dp"

android:paddingBottom="15dp">

<TextView

android:id="@+id/name\_of\_item"

android:layout\_height="wrap\_content"

android:layout\_width="match\_parent"

android:layout\_weight="1">

</TextView>

<ProgressBar

style="?android:attr/progressBarStyleHorizontal"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/progress\_item"

android:layout\_weight="1"

android:indeterminate="false"

android:max="100"

android:progress="0">

</ProgressBar>

<TextView

android:layout\_width="match\_parent"

android:gravity="right"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:id="@+id/number\_progress">

</TextView>

</LinearLayout>

В lab2\_activity.xml, в свою очередь, было добавлено два поля EditText для ввода заголовка и рейтинга и кнопка для добавления записей.

При написании кода в активити нужно было выполнить две цели: осуществить добавление записей и их восстановление при повороте экрана. Соответственно, сама функция вывода записей на экран разделяется на два возможных сценария: либо она просто выводит все записи, которые уже были сохранены, либо добавляет новую запись к существующим и сохраняет их.

В обоих случаях для этого используется LayoutInflater:

LayoutInflater inflater = getLayoutInflater();

При добавлении новой записи нужно получить введенные значения из EditBox’ов:

String nameOfItem = ((TextView)view.findViewById(R.id.edit\_name))

.getText().toString();

String ratingStr = ((TextView)view.findViewById(R.id.edit\_rating))

.getText().toString();

После проверки введенного значения рейтинга (это должно быть число от 0 до 10), заголовок и рейтинг добавляются в соответствующие списки.

namesOfItems.add(nameOfItem);

ratings.add(rating);

Затем запись добавляется на экран.

//подготовка добавления новой View посредством inflater’а

View newView = inflater.inflate(R.layout.rating\_item, linearLayout, false);

//задание заголовка в добавляемой записи

TextView nameOfItemView = newView.findViewById(R.id.name\_of\_item);

nameOfItemView.setText(nameOfItem);

//указание прогресса в виде полоски

ProgressBar progressBar = newView.findViewById(R.id.progress\_item);

progressBar.setProgress((int)rating\*10);

//указание прогресса в виде числа

TextView progressNum = newView.findViewById(R.id.number\_progress);

progressNum.setText(Double.toString(rating));

//добавление записи в отображаемый список

linearLayout.addView(newView);

В случае, если нужно вывести на экран уже существующие записи (что будет выполняться при повороте экрана), в качестве параметра функции передается номер записи в списках заголовков и рейтингов, в результате чего можно получить нужные элементы для отображения.

Для сохранения состояния при повороте экрана были перезаписаны функции onSaveInstanceState и onRestoreInstanceState.

В первой функции осуществляется передача информации о записях из списков в массивы, которые сохраняются.

String[] namesArray = new String[namesOfItems.size()];

double[] ratingsArray = new double[ratings.size()];

//передача данных из списков в массивы

for (int i = 0; i < namesArray.length; i++)

{

namesArray[i] = namesOfItems.get(i);

ratingsArray[i] = ratings.get(i);

}

//сохранение

outState.putStringArray("namesOfItems", namesArray);

outState.putDoubleArray("ratings", ratingsArray);

Во второй функции происходит обратное действие: данные из сохраненных массивов передаются в списки, которые в этом же цикле отображаются на экране.

for (int i = 0; i<nameArray.length; i++)

{

namesOfItems.add(nameArray[i]);

ratings.add(ratingsArray[i]);

AddNewItem(i);

}

# Работа приложения

Экран модуля данной лабораторной работы содержит два текстовых поля и кнопку для добавления записей, как можно видеть на рисунке 1.



Рисунок 1 – Экран модуля

После ввода названия и рейтинга и нажатия на кнопку происходит добавление записей, как показано на рисунках 2 и 3.

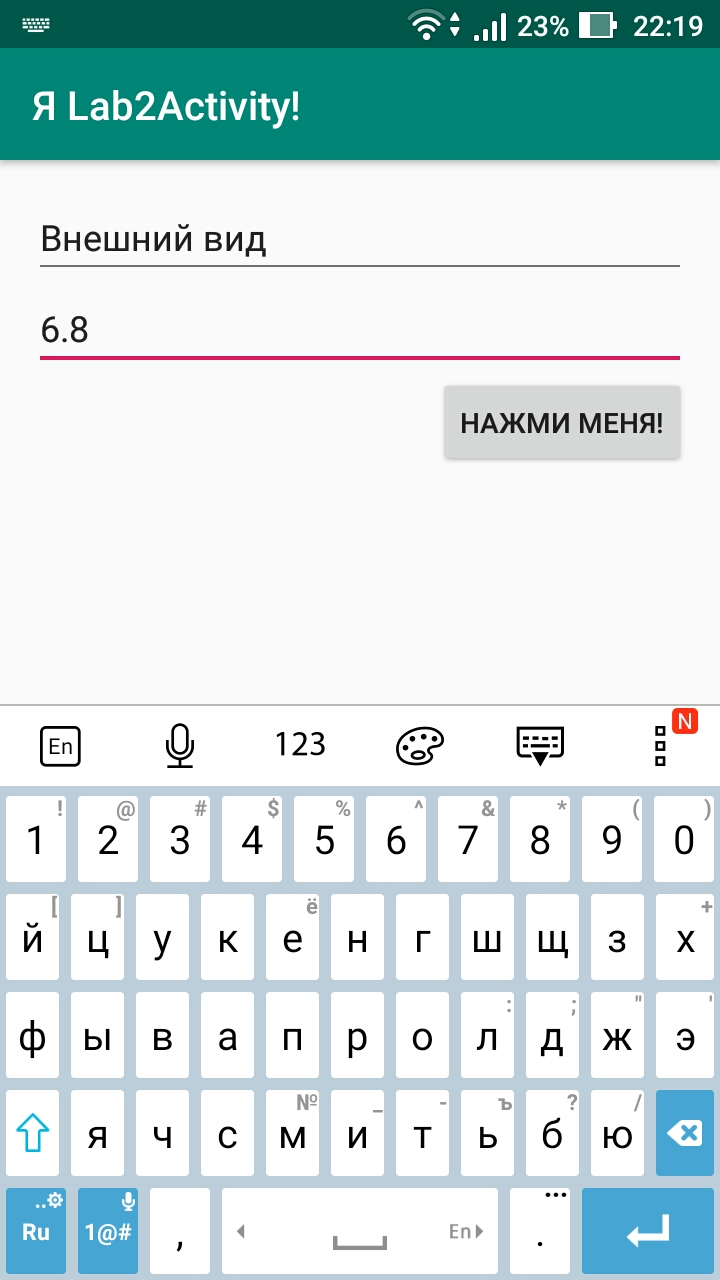


Рисунок 2 – Ввод заголовка и рейтинга

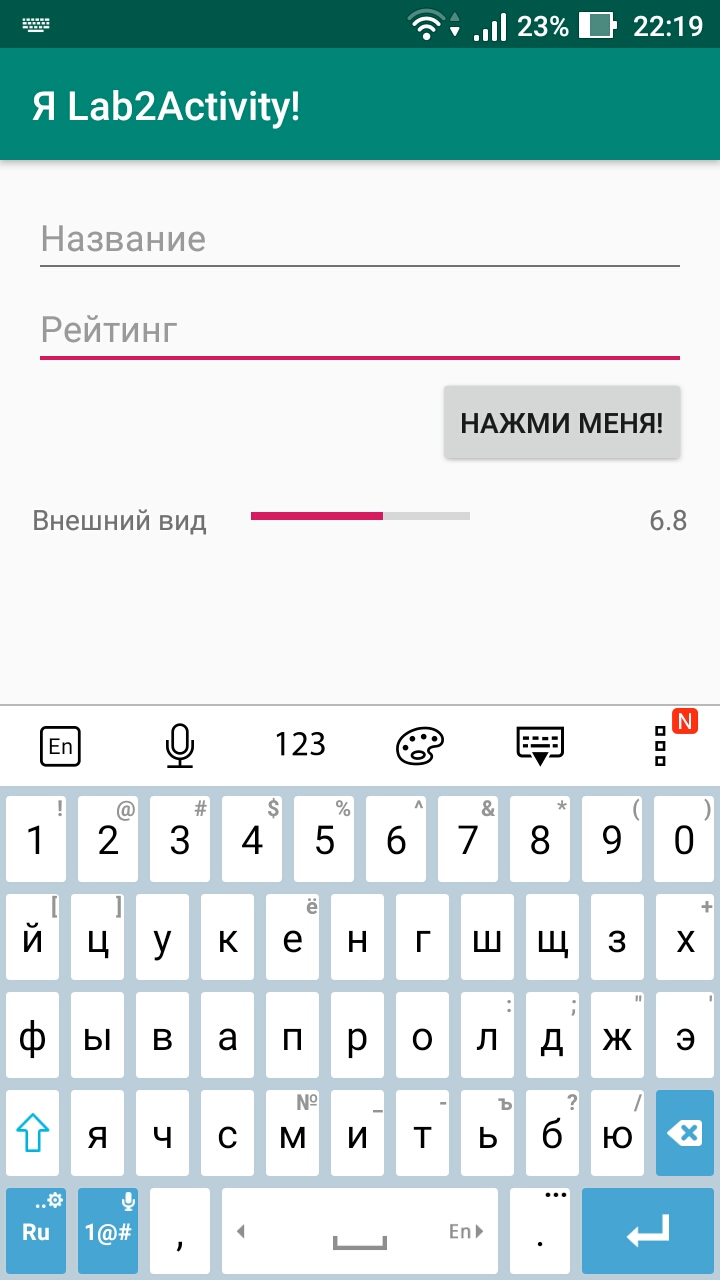


Рисунок 3 – Добавленная запись

При вводе в поле рейтинга нечисловых символов или чисел, меньших нуля или больших десяти, строка ввода очищается с появлением соответствующего сообщения, как можно видеть на рисунке 4.

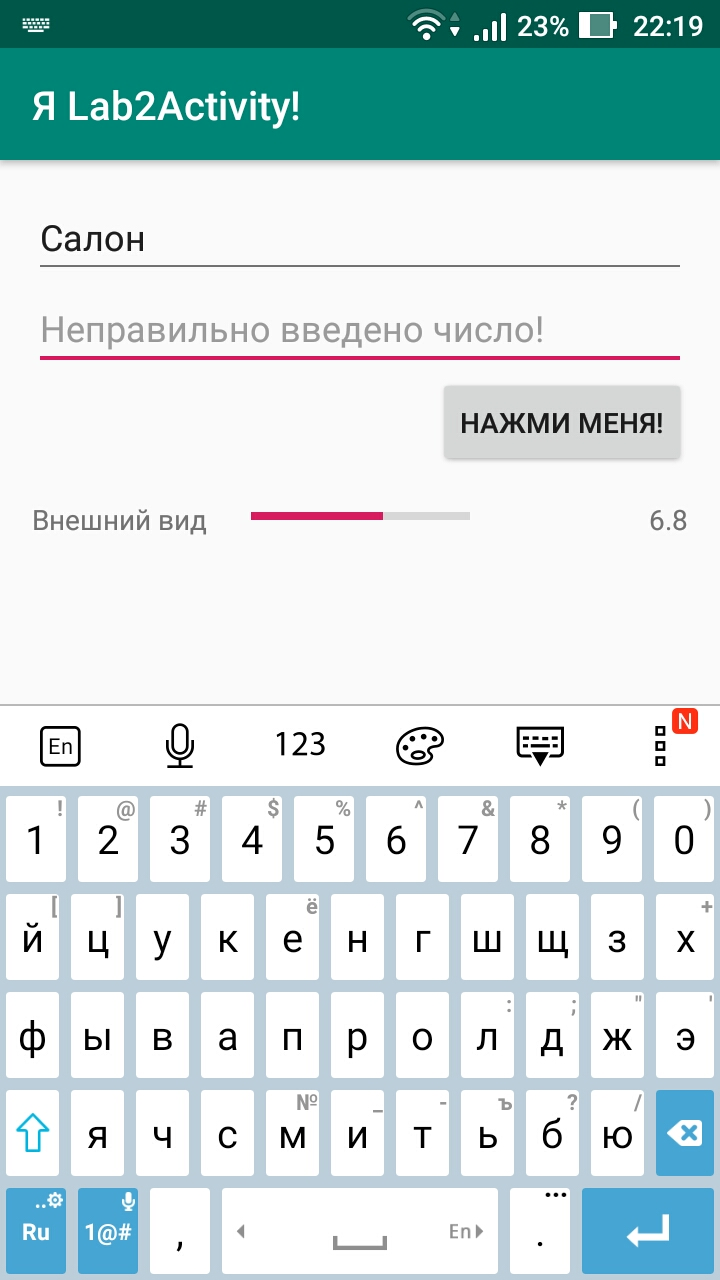


Рисунок 4 – Обработка неправильного ввода

Для проверки сохранения состояния были добавлены записи, показанные на рисунке 5.

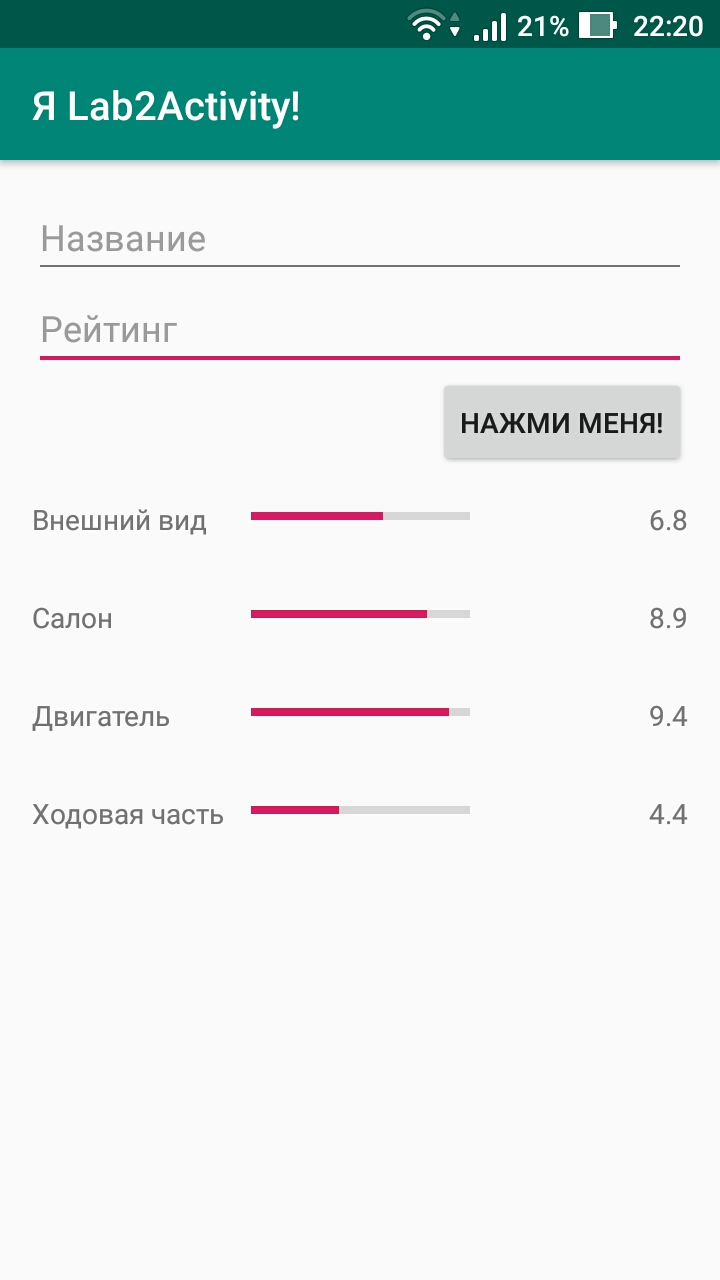


Рисунок 5 – Добавленные записи

При перевороте экрана записи не исчезают, так как информация о них сохраняется и после переворота экрана восстанавливается, как можно видеть на рисунке 6.

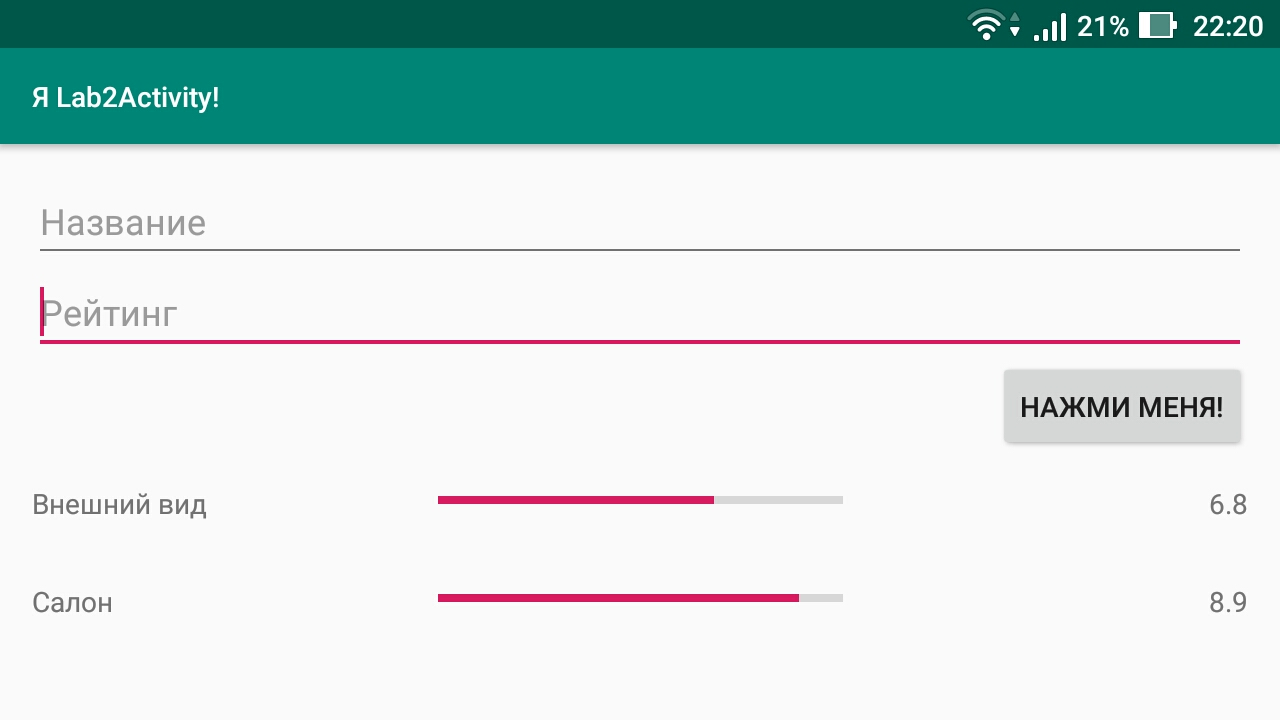


Рисунок 6 – Сохранение записей после переворота экрана

# Итоги работы

В результате выполнения лабораторной работы была реализована программа, в которой было необходимо научиться создавать и работать с кастомными View, а также сохранять состояние активити (например, на случай переворота экрана). Таким образом, здесь были изучены базовые методы создания пользовательского интерфейса приложения на ОС Android, при этом возможность сохранения состояния является не менее важным, так как оно позволяет избежать множества возможных неприятных ситуаций для пользователя (на примере данной лабораторной работы – без сохранения внесенных записей они бы стирались каждый раз, когда пользователь переворачивал бы экран (специально или случайно)).