



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных
технологий

Отчет по практической работе №4

по дисциплине «Разработка мобильных приложений»

Выполнили:

Студенты группы ИКБО-12-22

Солобай А.П.

Проверил:

Преподаватель

Степанов П.В.

Москва 2024 г.

Выполнение практической работы

Ссылка на GitHub со всеми работами: <https://github.com/Eckorezze/Mobil>

1. Контейнер LinearLayout. Вес элемента. Программное создание LinearLayout. Атрибут Layout_gravity.

Листинг 1 – Программное создание контейнера LinearLayout с весом элемента и атрибутом Layout_gravity.

```
LinearLayout linearLayout = new LinearLayout(this);
LinearLayout.LayoutParams layoutParams = new
LinearLayout.LayoutParams(
    LinearLayout.LayoutParams.MATCH_PARENT,
    LinearLayout.LayoutParams.MATCH_PARENT);
linearLayout.setOrientation(LinearLayout.VERTICAL);
layoutParams.gravity = Gravity.CENTER;

TextView textView1 = new TextView(this);
textView1.setText("Практика 4");
textView1.setLayoutParams(layoutParams);
textView1.setTextSize(14);

TextView textView2 = new TextView(this);
textView2.setText("Практика 5");
textView2.setLayoutParams(layoutParams);
textView2.setTextSize(14);

linearLayout.addView(textView1, new LinearLayout.LayoutParams
(LinearLayout.LayoutParams.MATCH_PARENT,
LinearLayout.LayoutParams.WRAP_CONTENT, 2));
linearLayout.addView(textView2, new LinearLayout.LayoutParams
(LinearLayout.LayoutParams.MATCH_PARENT,
LinearLayout.LayoutParams.WRAP_CONTENT, 1));

binding.toggleButton.setOnClickListener(v -> {

    Intent intent = new Intent(PracticeActivity.this,
TestActivity.class);
    startActivity(intent);
});

binding.toggleButton2.setOnClickListener(v -> {

    Intent intent = new Intent(PracticeActivity.this,
MainActivity.class);
    startActivity(intent);
});
```

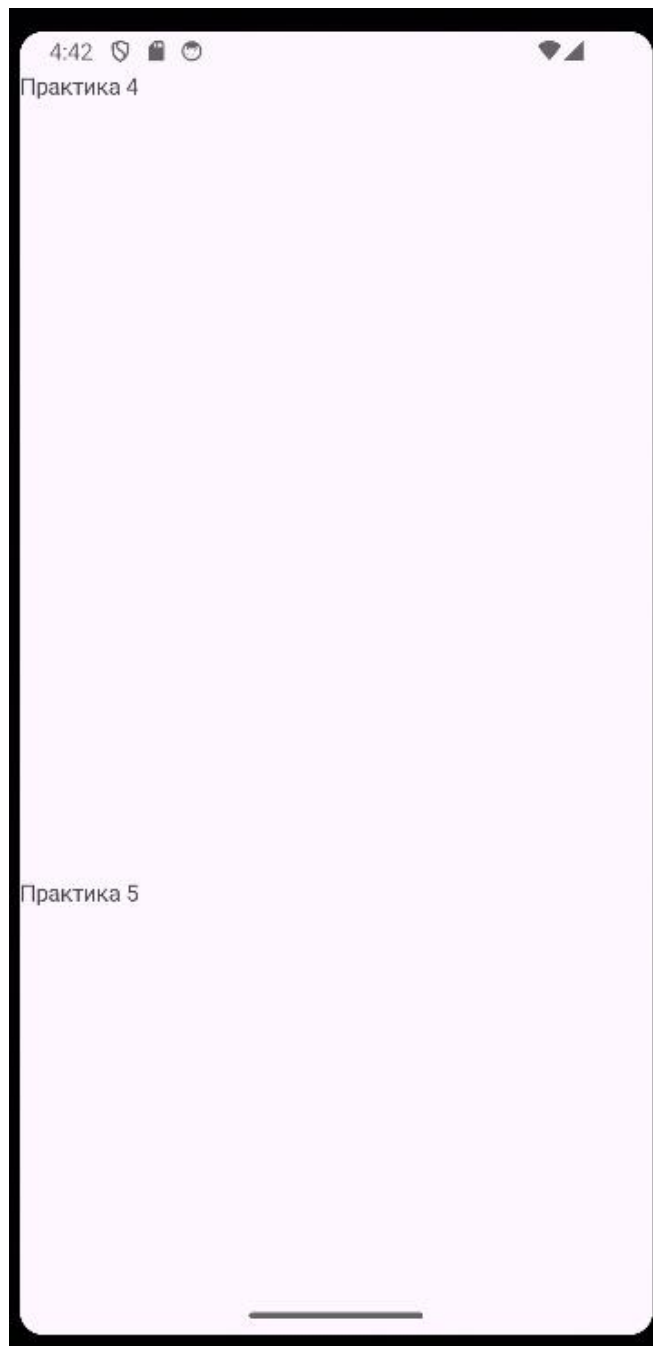


Рисунок 1 – тестирование кода для LinearLayout

2. Контейнер RelativeLayout (Legacy). Программное создание RelativeLayout.

Листинг 2 – Программное создание контейнера RelativeLayout.

```
RelativeLayout relativeLayout = new RelativeLayout(this);
RelativeLayout.LayoutParams textView = new
    RelativeLayout.LayoutParams(
        RelativeLayout.LayoutParams.MATCH_PARENT,
        RelativeLayout.LayoutParams.WRAP_CONTENT
    );
```

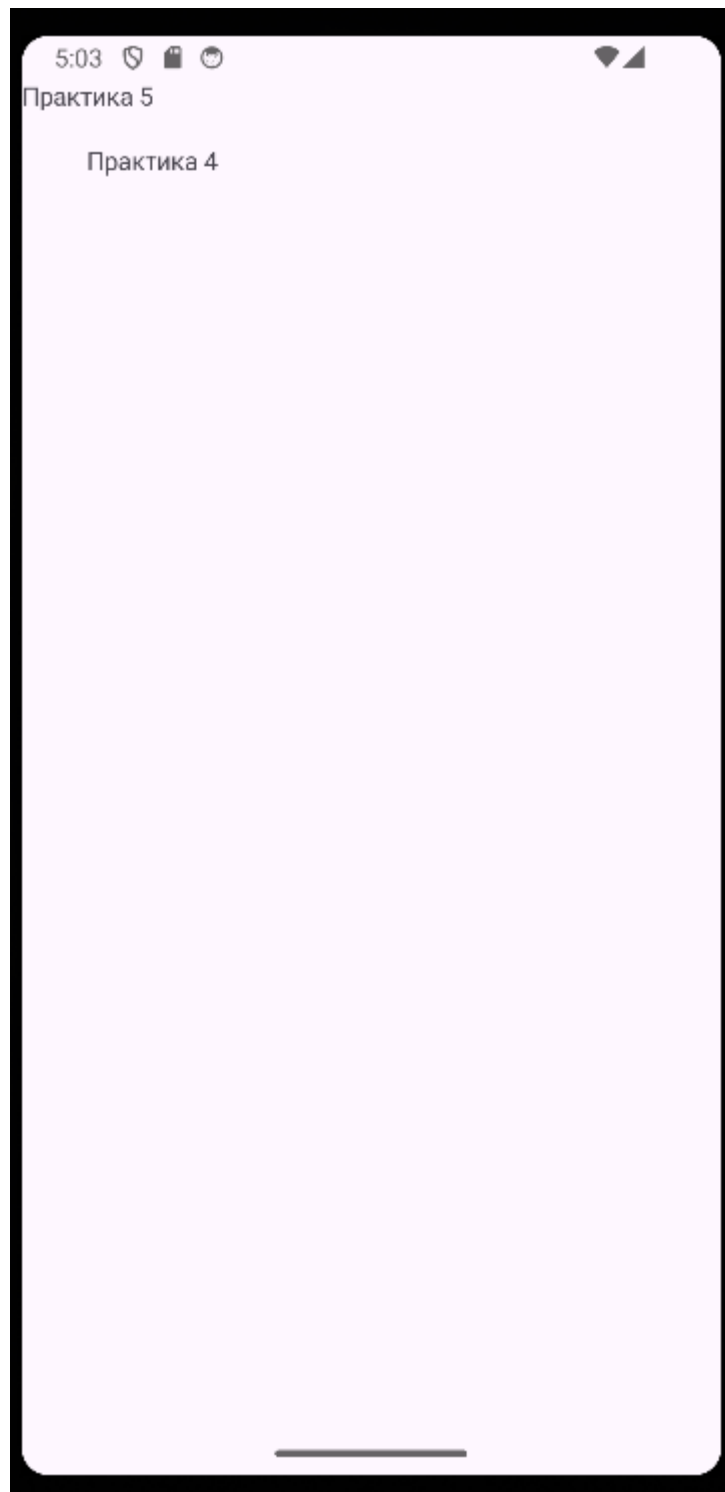


Рисунок 2 – тестирование кода для RelativeLayout

3. Контейнер TableLayout. Элемент TableRow. Атрибут layout_span.

Программное создание TableLayout.

Листинг 3 – Программное создание контейнера TableLayout с элементом TableRow, атрибутом layout_span.

```
TableLayout tableLayout = new TableLayout( this);

TableRow tableRow1 = new TableRow(this);
TextView textView1 = new TextView(this);
textView1.setText("Имя");
tableRow1.addView(textView1, new TableRow.LayoutParams(
    TableRow.LayoutParams.WRAP_CONTENT,
    TableRow.LayoutParams.WRAP_CONTENT, 0.5f));
EditText editText1 = new EditText(this);
tableRow1.addView(editText1, new TableRow.LayoutParams(
    TableRow.LayoutParams.WRAP_CONTENT,
    TableRow.LayoutParams.WRAP_CONTENT, 1.0f));
// вторая строка
TableRow tableRow2 = new TableRow(this);
TextView textView2 = new TextView(this);
textView2.setText("Фамилия");
tableRow2.addView(textView2, new TableRow.LayoutParams(
    TableRow.LayoutParams.WRAP_CONTENT,
    TableRow.LayoutParams.WRAP_CONTENT, 0.5f));
EditText editText2 = new EditText(this);
tableRow2.addView(editText2, new TableRow.LayoutParams(
    TableRow.LayoutParams.WRAP_CONTENT,
    TableRow.LayoutParams.WRAP_CONTENT, 1.f));
tableLayout.addView(tableRow1);
tableLayout.addView(tableRow2);
```

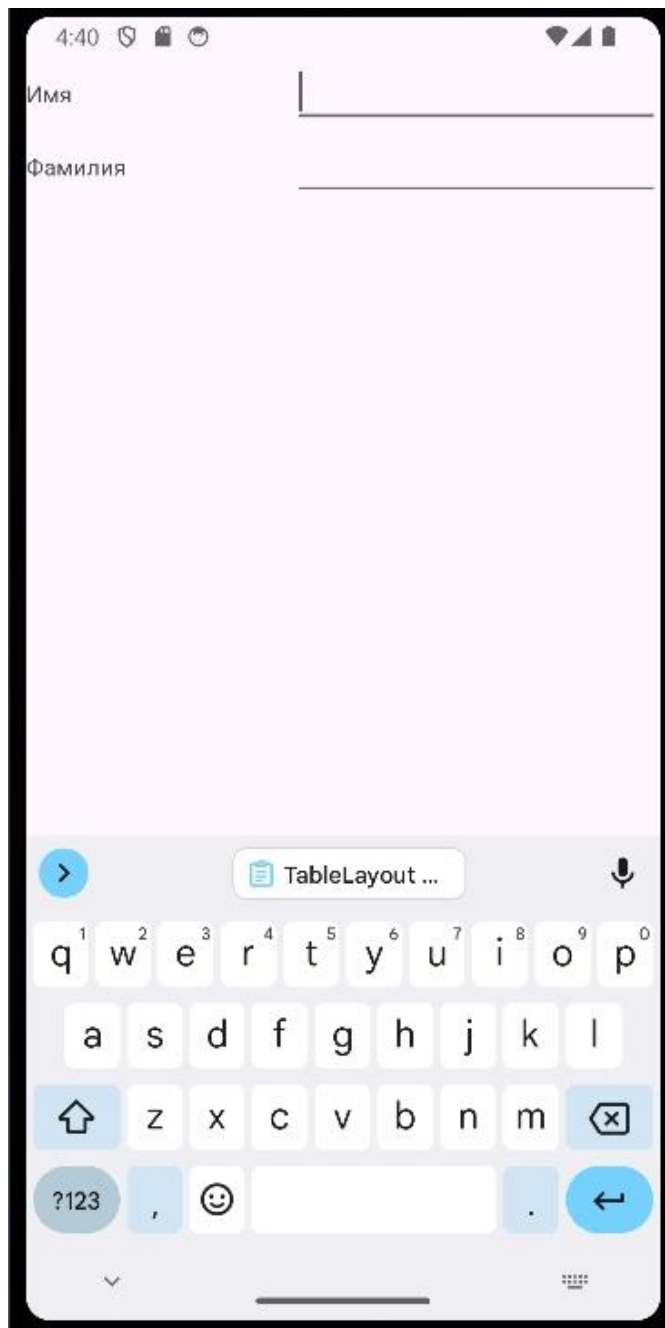


Рисунок 3 – тестирование кода для TableLayout

4. Контейнер FrameLayout. Атрибут android:layout_gravity. Программное создание FrameLayout.

Листинг 4 – Программное создание контейнера TableLayout с элементом TableRow, атрибутом layout_span.

```
FrameLayout frameLayout = new FrameLayout(this);  
  
TextView textView1 = new TextView(this);  
textView1.setText("Практика 4");  
textView1.setTextSize(14);
```

```

FrameLayout.LayoutParams layoutParams = new
    FrameLayout.LayoutParams
        (FrameLayout.LayoutParams.WRAP_CONTENT,
            FrameLayout.LayoutParams.WRAP_CONTENT);
layoutParams.gravity = Gravity.CENTER_HORIZONTAL | Gravity.TOP;
textView1.setLayoutParams(layoutParams);
textView1.setTextSize(26);
frameLayout.addView(textView1);

```



Рисунок 4 – тестирование кода для TableLayout

5. Контейнер GridLayout. Атрибуты android:rowCount и android:columnCount. Атрибуты android:layout_column, android:layout_row, android:layout_columnSpan, android:layout_rowSpan. Программное создание GridLayout. Класс GridLayout.LayoutParams. Свойство columnSpec, rowSpec, leftMargin, rightMargin, topMargin, bottomMargin, width, height. Объект GridLayout.Spec.

Листинг 5 – Программное создание контейнера GridLayout с требуемыми атрибутами и свойствами.

```
GridLayout gridLayout = new GridLayout( this);
Button btn = new Button(this);
btn.setText("click");
GridLayout.LayoutParams layoutParams = new GridLayout.LayoutParams();
layoutParams.columnSpec = GridLayout.spec(0,2);
layoutParams.rowSpec = GridLayout.spec(1,1);
layoutParams.leftMargin=5;
layoutParams.rightMargin=5;
layoutParams.topMargin=4;
layoutParams.bottomMargin=4;
layoutParams.width = GridLayout.LayoutParams.MATCH_PARENT;
layoutParams.height = GridLayout.LayoutParams.WRAP_CONTENT;
gridLayout.addView(btn, layoutParams);
```

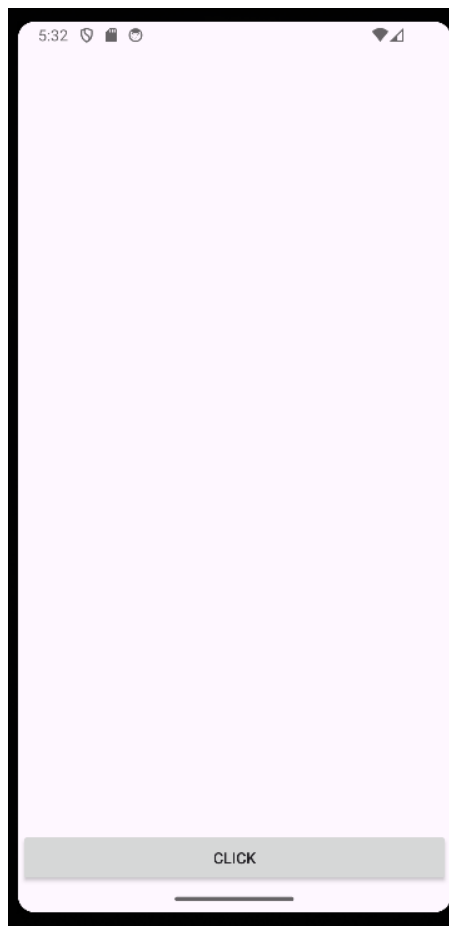


Рисунок 5 – тестирование кода для GridLayout

6. Контейнер ScrollView. Создание ScrollView программно в коде MainActivity.

Листинг 6 – Программное создание контейнера ScrollView

```
ScrollView scrollView = new ScrollView(this);
TextView textView = new TextView(this);
textView.setText("Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Morbi ac diam blandit sem laoreet blandit id eget
leo. Nam blandit tellus eget tempor eleifend. Phasellus id varius
ipsum, ac sagittis risus. Morbi dignissim mollis ligula eget rhoncus.
Maecenas bibendum turpis erat, non sollicitudin enim faucibus nec.
Duis volutpat felis et enim ornare, eu pretium urna efficitur. Nulla
facilisi. Quisque elementum faucibus ullamcorper. Vivamus blandit,
magna et sodales tempor, quam turpis sollicitudin mauris, eu
sollicitudin magna nibh eget tellus. Vivamus sit amet eleifend
odio.\n" +
    "\n" +
    "Fusce tincidunt posuere nisl hendrerit mollis. Cras
rutrum congue nunc a gravida. Vestibulum ante ipsum primis in
faucibus orci luctus et ultrices posuere cubilia curae; Aliquam non
hendrerit enim. Nullam mi lectus, posuere eu rutrum ac, vehicula
vitae magna. Sed ut augue sed est faucibus lobortis in non nisi.
Phasellus aliquet turpis vitae faucibus consectetur.\n" +
    "\n" +
    "Vestibulum ante nisi, molestie sed bibendum sit amet,
egestas faucibus ipsum. Integer non congue nulla, sed congue metus.
Pellentesque hendrerit ex et ligula accumsan, et suscipit magna
pharetra. Quisque fermentum lorem ac consectetur lobortis. Aenean
volutpat lorem non lacus pulvinar, in commodo ipsum dignissim. Etiam
mollis lacus quis dignissim dignissim. Nulla in consequat libero.
Integer sem mauris, ullamcorper eget consequat id, laoreet vel nisl.
Etiam gravida sollicitudin bibendum. Cras bibendum purus nibh, et
luctus erat aliquet nec. Ut sit amet neque vel velit vulputate porta
id ac erat. Aenean lorem enim, condimentum ut justo eget,
sollicitudin efficitur nisl. Ut molestie, lorem vitae tempor
condimentum, nisi sem tristique libero, non porta lectus eros a
felis. Sed sit amet elit velit. Integer vulputate commodo sapien, eu
dignissim magna sollicitudin in.\n" +
    "\n" +
    "In sollicitudin lacinia luctus. Sed et ipsum
ullamcorper, tristique odio eget, dictum augue. Mauris at hendrerit
odio. Nullam magna odio, lobortis sit amet efficitur at, luctus sed
diam. Nunc volutpat placerat elit quis lacinia. Sed non pharetra ex.
Nunc mattis, neque nec finibus aliquam, sem metus finibus nibh, sed
finibus ipsum lectus eu libero. Suspendisse potenti. Morbi dignissim
vehicula nisi vitae consectetur. Phasellus suscipit enim vitae erat
maximus, non eleifend diam ullamcorper. Phasellus orci ante,
vestibulum a odio id, bibendum iaculis mi. Maecenas sagittis ac enim
et bibendum.\n" +
    "\n" +
    "Vivamus rhoncus, enim accumsan faucibus gravida, elit
elit pulvinar enim, eleifend aliquam lectus erat vel lacus. Nulla sit
amet scelerisque quam. Quisque mollis mauris at diam auctor
vestibulum et et quam. Nullam fermentum dui facilisis efficitur
egestas. Maecenas ac arcu tincidunt, fermentum purus non, euismod
dolor. Aliquam erat volutpat. Phasellus aliquam purus massa, in
```

```

cursus lacus sollicitudin a. Ut eget lectus nulla.");
textView.setLayoutParams(new ViewGroup.LayoutParams
(ViewGroup.LayoutParams.WRAP_CONTENT,
ViewGroup.LayoutParams.WRAP_CONTENT));
textView.setTextSize(26);
scrollView.addView(textView);
setContentView(scrollView);
}

```

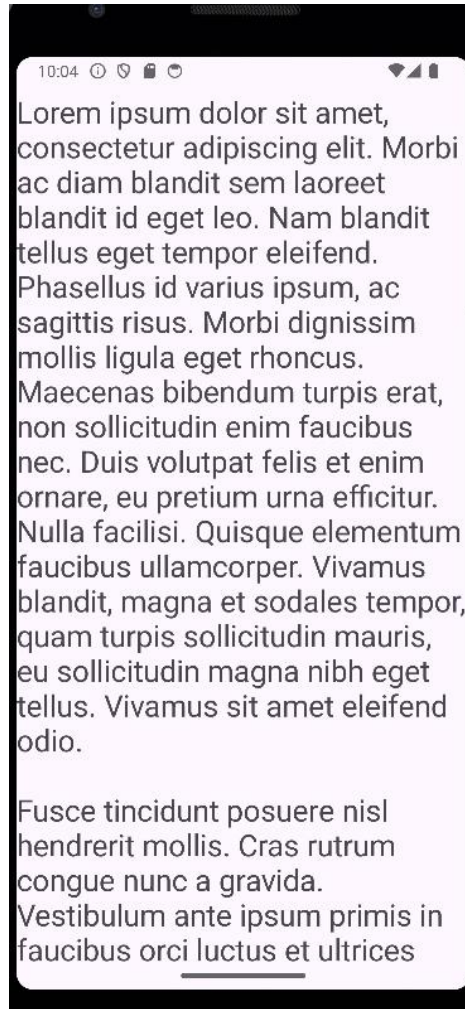


Рисунок 6 – тестирование кода для ScrollView

7. Вложенные layout. Элемент include.

Листинг 7 – Создание вложенности через include

```
<include  
    android:id="@+id/button"  
    layout="@layout/button_panel"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    app:layout_constraintLeft_toLeftOf="parent"  
>
```

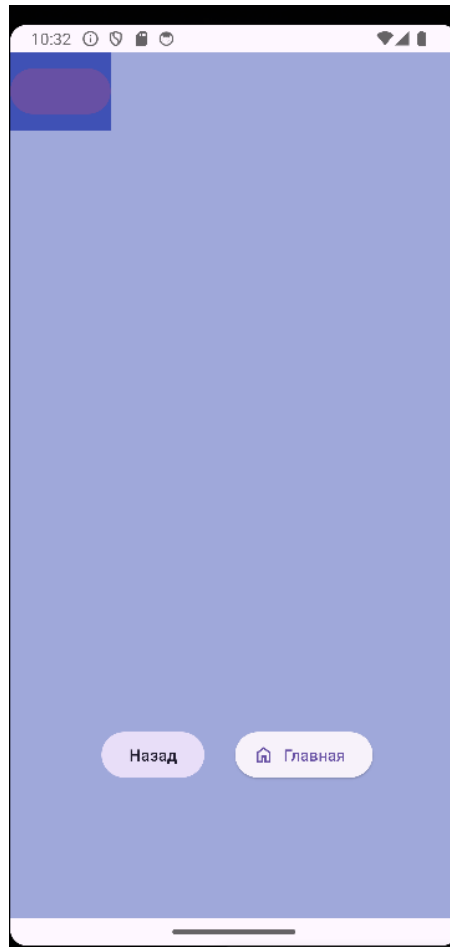


Рисунок 7 – тестирование кода для include

Вывод

В ходе работы мне удалось реализовать различные разметки с помощью кода на Java. Получены навыки создания ScrollView активности, а также использования вложенности через include.