|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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 группыИКБО-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); }); |

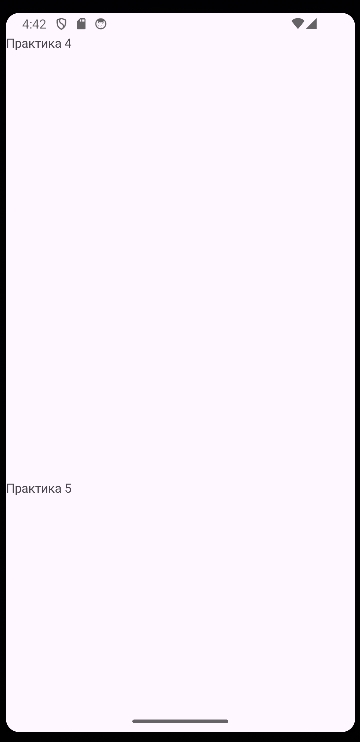


Рисунок 1 – тестирование кода для LinearLayout

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 ); |

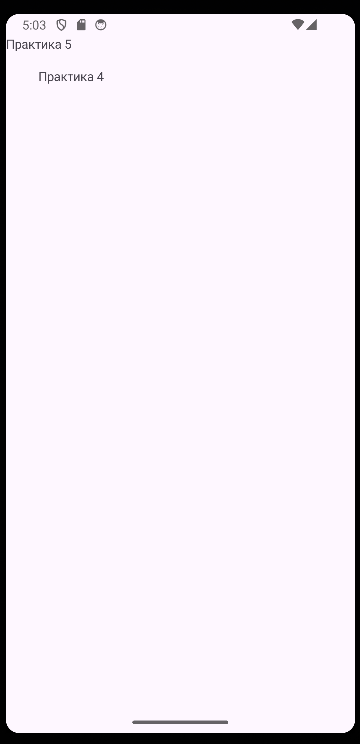


Рисунок 2 – тестирование кода для RelativeLayout

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); |

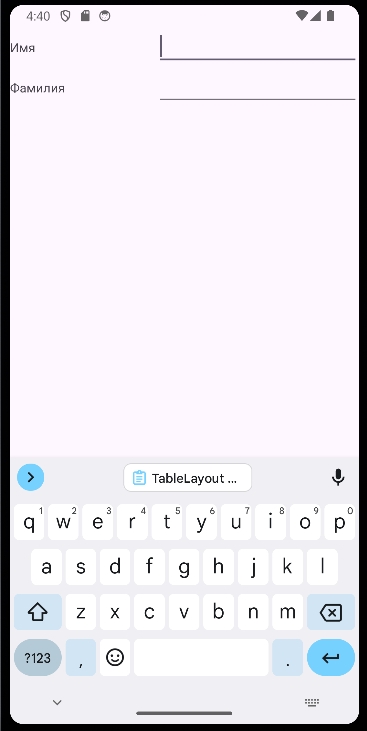


Рисунок 3 – тестирование кода для TableLayout

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); |

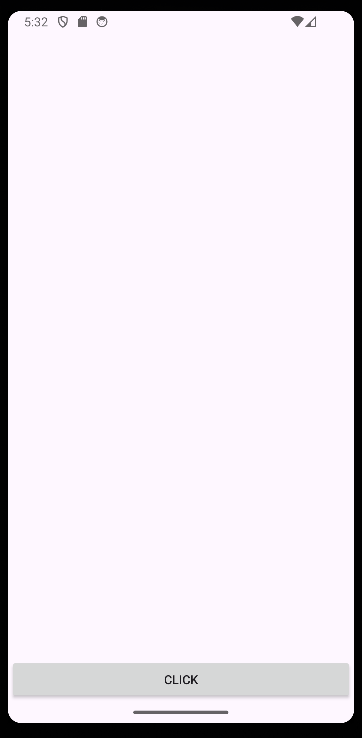


Рисунок 5 – тестирование кода для GridLayout

# Вывод

В ходе работы мне удалось реализовать элементы интерфейса с помощью кода на Java. Получены навыки реализации получения и управления визуальными элементами в коде на Java.