



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 №5**

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. Элемент TextView и его атрибуты. Установка элемента в коде. Используя атрибут android:autoLink вывести на экран ссылку и телефон.

*Листинг 1 – программная реализация элемента TextView с необходимыми требованиями*

```
ConstraintLayout constraintLayout = new ConstraintLayout(this);
TextView textView = new TextView(this);
textView.setAllCaps(false);
textView.setText("https://vk.com/im");
textView.setTypeface(Typeface.create("casual", Typeface.BOLD));
textView.setTextSize(20);
textView.setHeight(ViewGroup.LayoutParams.WRAP_CONTENT);
textView.setWidth(ViewGroup.LayoutParams.WRAP_CONTENT);
Linkify.addLinks(textView, Linkify.WEB_URLS);
TextView textView2 = new TextView(this);
textView2.setAllCaps(true);
textView2.setText("+78005553535");
textView2.setPadding(0, 200, 0, 0);
textView2.setTypeface(Typeface.create("casual", Typeface.BOLD));
textView2.setTextSize(20);
Linkify.addLinks(textView2, Linkify.PHONE_NUMBERS);
textView.setLinksClickable(true);
ConstraintLayout.LayoutParams layoutParams = new
    ConstraintLayout.LayoutParams
        (ConstraintLayout.LayoutParams.MATCH_PARENT,
            ConstraintLayout.LayoutParams.MATCH_PARENT);
textView.setLayoutParams(layoutParams);
textView2.setLayoutParams(layoutParams);
constraintLayout.addView(textView);
constraintLayout.addView(textView2);
```



Рисунок 1 – тестирование кода для TextView с почтой и номером телефона

2. Элемент EditText и его атрибуты. Используя атрибуты `android:hint` и `android:inputType` задать текст, который будет отображаться в качестве подсказки, если элемент EditText пуст и клавиатуру для ввода. Реализовать два поля. Первое поле - однострочное, а второе – многострочное. Введенные символы в первом поле – отображаются во втором.

*Листинг 2 – программная реализация EditText с необходимыми требованиями.*

```
LinearLayout linearLayout = new LinearLayout(this);
linearLayout.setOrientation(LinearLayout.VERTICAL);
LinearLayout.LayoutParams layoutParams = new
LinearLayout.LayoutParams(
    LinearLayout.LayoutParams.MATCH_PARENT,
    LinearLayout.LayoutParams.WRAP_CONTENT
);
EditText editText1 = new EditText(this);
EditText editText2 = new EditText(this);

editText1.setHint("Введите ваше имя");
editText1.setInputType(android.text.InputType.TYPE_CLASS_TEXT);
```

```

editText1.addTextChangedListener(new TextWatcher() {

    @Override
    public void beforeTextChanged(CharSequence s, int start, int
count, int after) {

    }

    @Override
    public void onTextChanged(CharSequence s, int start, int before,
int count) {
        editText2.setText(s.toString());
    }

    @Override
    public void afterTextChanged(Editable s) {

    }

});

editText1.setLayoutParams(layoutParams);
editText2.setLayoutParams(layoutParams);
linearLayout.addView(editText1);
linearLayout.addView(editText2);

```

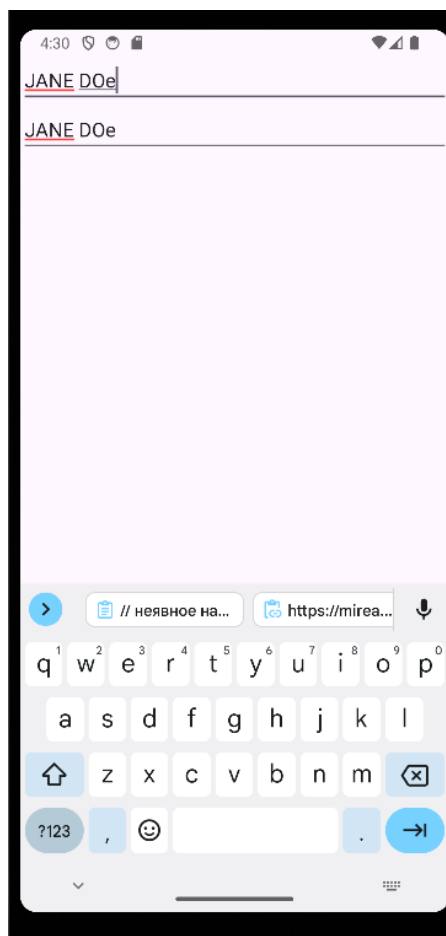


Рисунок 2 – тестирование кода для EditText

3. Элемент Button и его атрибуты. Реализовать на экране кнопку с надписью “Ввод”. После нажатия на кнопку выводится текст из первого поля во второе. Реализовать аналогичный пример полностью в коде.

*Листинг 3 – программная реализация Button с необходимыми требованиями.*

```
ConstraintLayout constraintLayout = new ConstraintLayout(this);

TextView textView = new TextView(this);
EditText editText = new EditText(this);
Button button = new Button(this);
button.setText("Ввод");

editText.setHint("Введите ваше имя");
editText.setInputType(android.text.InputType.TYPE_CLASS_TEXT);

ConstraintLayout.LayoutParams buttonLayout = new
ConstraintLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT,
    ViewGroup.LayoutParams.WRAP_CONTENT);

buttonLayout.topToTop = ConstraintLayout.LayoutParams.PARENT_ID;
buttonLayout.bottomToBottom =
ConstraintLayout.LayoutParams.PARENT_ID;
buttonLayout.startToStart = ConstraintLayout.LayoutParams.PARENT_ID;
buttonLayout.endToEnd = ConstraintLayout.LayoutParams.PARENT_ID;
button.setLayoutParams(buttonLayout);
constraintLayout.addView(button);

ConstraintLayout.LayoutParams editTextLayout = new
    ConstraintLayout.LayoutParams(
        ConstraintLayout.LayoutParams.MATCH_CONSTRAINT,
        ConstraintLayout.LayoutParams.WRAP_CONTENT
    );
editTextLayout.topToBottom = textView.getId();
editTextLayout.leftToLeft = ConstraintLayout.LayoutParams.PARENT_ID;
editTextLayout.rightToRight =
    ConstraintLayout.LayoutParams.PARENT_ID;
editText.setLayoutParams(editTextLayout);
constraintLayout.addView(editText);

ConstraintLayout.LayoutParams textViewLayout = new
    ConstraintLayout.LayoutParams(
        ConstraintLayout.LayoutParams.MATCH_CONSTRAINT,
        ConstraintLayout.LayoutParams.WRAP_CONTENT
    );
textViewLayout.topToBottom = editText.getId();
textViewLayout.leftToLeft = ConstraintLayout.LayoutParams.PARENT_ID;
textViewLayout.rightToRight =
    ConstraintLayout.LayoutParams.PARENT_ID;
textViewLayout.bottomToBottom =
    ConstraintLayout.LayoutParams.PARENT_ID;
textViewLayout.bottomMargin = 1500;
textViewLayout.width = ConstraintLayout.LayoutParams.WRAP_CONTENT;
textViewLayout.height = ConstraintLayout.LayoutParams.WRAP_CONTENT;

textView.setLayoutParams(textViewLayout);
```

```
constraintLayout.addView(textView);

button.setOnClickListener(v -> {
    textView.setText("Привет, " + editText.getText());
});
```

*Листинг 4 – реализация Button с необходимыми требованиями.*

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="160dp"
    android:layout_marginEnd="160dp"
    android:layout_marginBottom="435dp"
    android:text="Ввод"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

<EditText
    android:id="@+id/editText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="63dp"
    android:hint="Введите имя"
    android:minHeight="48dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView" />
```

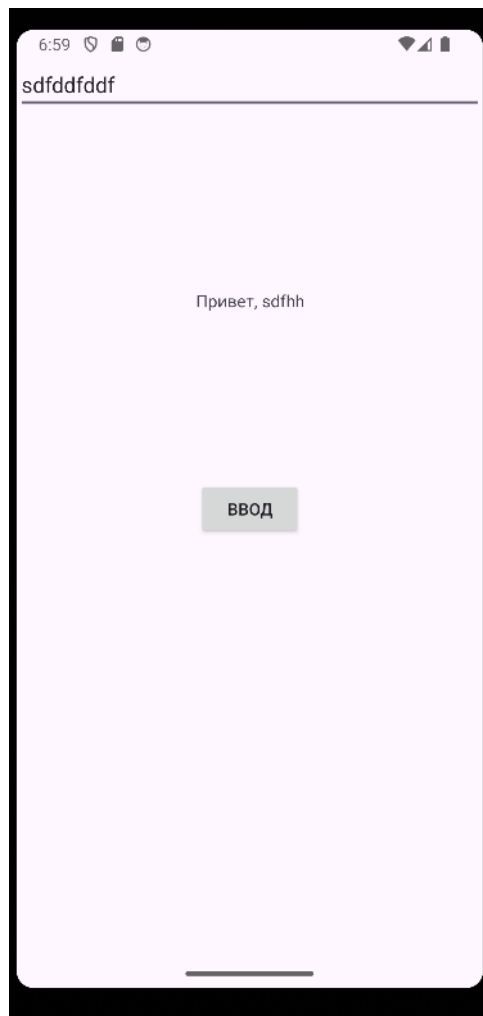


Рисунок 3 – тестирование кода для Button



Рисунок 4 – тестирование xml для Button

4. Класс Toast. Всплывающие окна. Toast можно использовать только в коде java. Реализуйте это, используя метод `Toast.makeText()`. В качестве времени показа окна можете использовать целочисленное значение - количество миллисекунд или встроенные константы `Toast.LENGTH_LONG` (2000 миллисекунд) и `Toast.LENGTH_SHORT` (2000 миллисекунд). Используйте метод `setGravity` для указания, в какой части контейнера надо позиционировать Toast.

*Листинг 5 – программная реализация Toast с необходимыми требованиями.*

```
binding.button.setOnClickListener(v -> {  
    Toast toast = Toast.makeText(this, "Практика 5",  
    Toast.LENGTH_LONG);  
    toast.setGravity(Gravity.TOP, 0,160); // import  
    android.view.Gravity;  
    toast.show();  
});
```



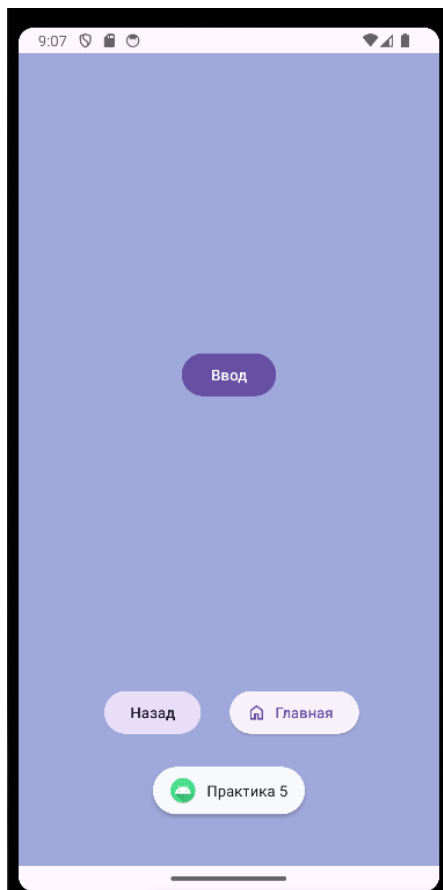


Рисунок 5 – тестирование кода для Toast с необходимыми параметрами

5. Элемент Snackbar. Реализуйте пример с помощью метода `make()`. Прикрепление обработчика события. Реализуйте пример с помощью метода `setAction()`. Реализуйте пример с настройкой визуального вида.

*Листинг 6 – программная реализация Snackbar с необходимыми требованиями.*

```
binding.button.setOnClickListener(v -> {
    Snackbar snackbar = Snackbar.make(view, "Практика 5",
        Snackbar.LENGTH_LONG);
    snackbar.setTextColor(0xFF81C784);
    snackbar.setBackgroundTint(0xFF555555);
    snackbar.setActionTextColor(0xFF0277BD);
    snackbar.setAction("Далее...", v1 -> {
        Toast toast = Toast.makeText(getApplicationContext(), "Далее
        clicked!", Toast.LENGTH_LONG);
        toast.show();
    });
    snackbar.show();
});
```

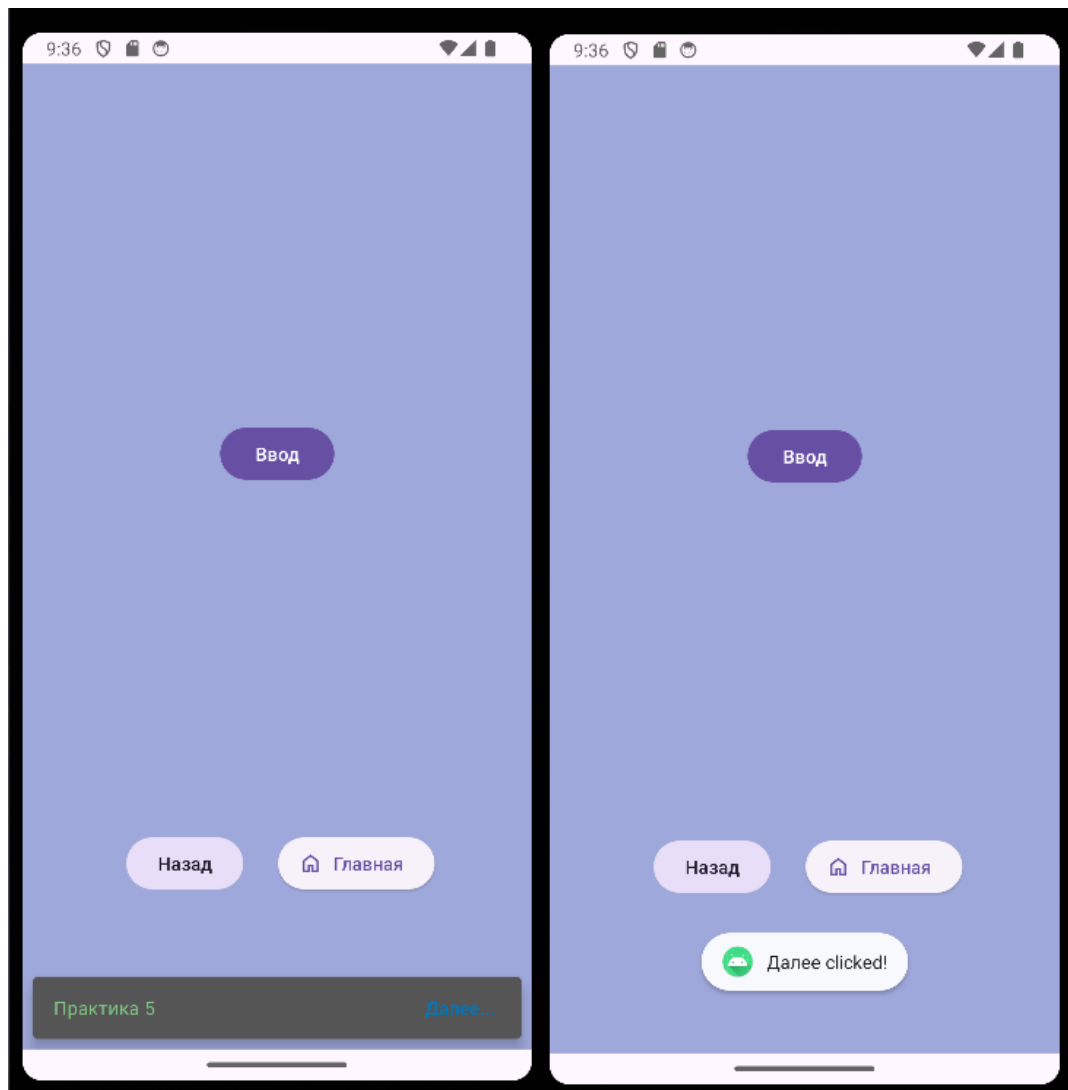


Рисунок 6,7 – тестирование кода для Snackbar

6. Элементы Checkbox. Реализуйте пример с несколькими флажками которые могут находиться в отмеченном и неотмеченном состоянии.

*Листинг 7 –реализация Checkbox.*

```
<TextView android:id="@+id/selection"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="26sp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent"/>
<CheckBox android:id="@+id/java"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Java"
    android:textSize="26sp"
    android:onClick="onCheckboxClicked"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/selection"/>
<CheckBox android:id="@+id/kotlin"
    android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:text="Kotlin"
android:textSize="26sp"
android:onClick="onCheckboxClicked"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintTop_toBottomOf="@+id/java"/>
```

*Листинг 8 – работа с Checkbox в коде на Java.*

```
public void onCheckboxClicked(View view) {
    CheckBox checkBox = (CheckBox) view;
    boolean checked = checkBox.isChecked();
    TextView selection = findViewById(R.id.selection);
    if (view.getId() == R.id.java)
        if (checked)
            Toast.makeText(this, "Вы выбрали Java",
                Toast.LENGTH_LONG).show();
    else if (view.getId() == R.id.kotlin)
        if (checked)
            Toast.makeText(this, "Вы выбрали Kotlin",
                Toast.LENGTH_LONG).show();
}
```



Рисунок 8 – тестирование кода для Checkbox

7. Слушатель `OnCheckedChangeListener`. Реализуйте пример с помощью метода `onCheckedChanged`.

*Листинг 9 – реализация `OnCheckedChangeListener`.*

```
TextView selection = findViewById(R.id.selection);
CheckBox enableBox = findViewById(R.id.enabled);
enableBox.setOnCheckedChangeListener((buttonView, isChecked) -> {
    if (isChecked) {
        selection.setText("Включено");
        buttonView.setText("Выключить");
    }
    else {
        selection.setText("Выключено");
        buttonView.setText("Включить");
    }
});
```



Рисунок 9 – тестирование кода для `OnCheckedChangeListener`

8. Элемент `ToggleButton`. Реализуйте пример с помощью атрибутов `android:textOn` и `android:textOff`. Реализуйте пример создания элемента `ToggleButton` в коде java.

*Листинг 10 – реализация `ToggleButton`.*

```
<ToggleButton
    android:id="@+id/toggle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="156dp"
    android:layout_marginTop="144dp"
    android:onClick="onToggleClicked"
    android:textOff="Выключено"
    android:textOn="Включено"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

*Листинг 11 – реализация `ToggleButton` в коде на Java.*

```
ConstraintLayout layout = new ConstraintLayout(this);
ConstraintLayout.LayoutParams layoutParams = new
    ConstraintLayout.LayoutParams
        (ConstraintLayout.LayoutParams.WRAP_CONTENT,
         ConstraintLayout.LayoutParams.WRAP_CONTENT);
ToggleButton toggleButton = new ToggleButton(this);
toggleButton.setTextOff("Выключено");
toggleButton.setTextOn("Включено");
toggleButton.setText("Выключено");
toggleButton.setOnClickListener(view1 -> {
    boolean on = ((ToggleButton) view1).isChecked();
    if (on) {
        Toast.makeText(getApplicationContext(), "Свет включен",
            Toast.LENGTH_LONG).show();
    } else {
        Toast.makeText(getApplicationContext(), "Свет выключен!",
            Toast.LENGTH_LONG).show();
    }
});
layoutParams.leftToLeft = ConstraintLayout.LayoutParams.PARENT_ID;
layoutParams.topToTop = ConstraintLayout.LayoutParams.PARENT_ID;
layout.addView(toggleButton);
```

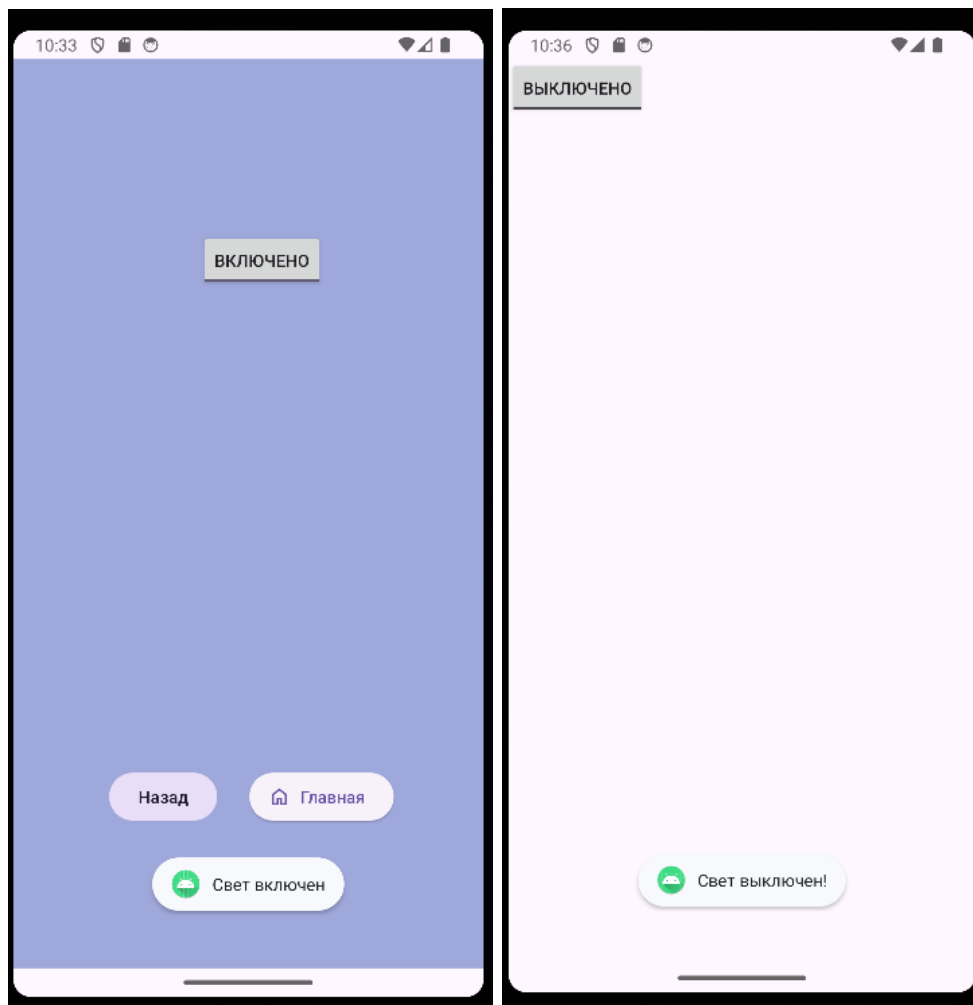


Рисунок 10,11 – тестирование кода для ToggleButton

## 9. Класс RadioButton Реализуйте пример.

Листинг 12 – реализация RadioButton.

```
<RadioGroup
    android:layout_width="417dp"
    android:layout_height="216dp"
    android:layout_marginStart="118dp"
    android:layout_marginTop="104dp"
    android:layout_marginEnd="120dp"
    android:layout_marginBottom="627dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.454"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0">

    <RadioButton
        android:id="@+id/kotlin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="onRadioButtonClicked"
        android:text="Kotlin" />
```

```
<RadioButton
    android:id="@+id/java"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="onRadioButtonClicked"
    android:text="Java" />
</RadioGroup>
```

*Листинг 13 – работа с RadioButton в коде на Java.*

```
public void onRadioButtonClicked(View view) {
    boolean checked = ((RadioButton) view).isChecked();
    TextView selection = binding.selection;
    if (view.getId() == R.id.java) {
        if (checked) {
            selection.setText("Выбрана Java");
        }
    } else if (view.getId() == R.id.kotlin) {
        if (checked) {
            selection.setText("Выбран Kotlin");
        }
    }
}
```

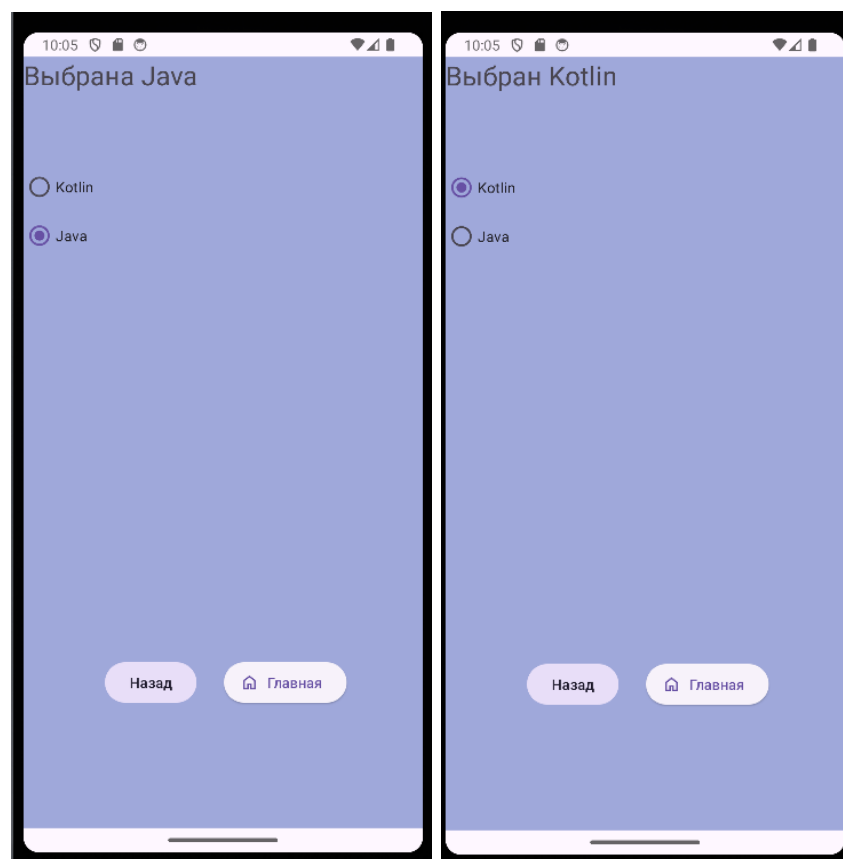


Рисунок 12,13 – тестирование кода для RadioButton

## 10. Элемент DatePicker. Реализуйте пример.

*Листинг 14 – реализация DatePicker.*

```
<DatePicker
    android:id="@+id/datePicker"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="30dp"
    android:layout_marginEnd="25dp"
    android:layout_marginBottom="176dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintLeft_toLeftOf="parent" />
```

*Листинг 15 – работа с DatePicker в коде на Java.*

```
TextView dateTextView = binding.dateTextView;
DatePicker datePicker = binding.datePicker;
datePicker.init(2024, 02, 23,
    (view1, year, monthOfYear, dayOfMonth) ->
dateTextView.setText("Дата: " + view1.getDayOfMonth() + "/" +
    (view1.getMonth() + 1) + "/" + view1.getYear()));
```

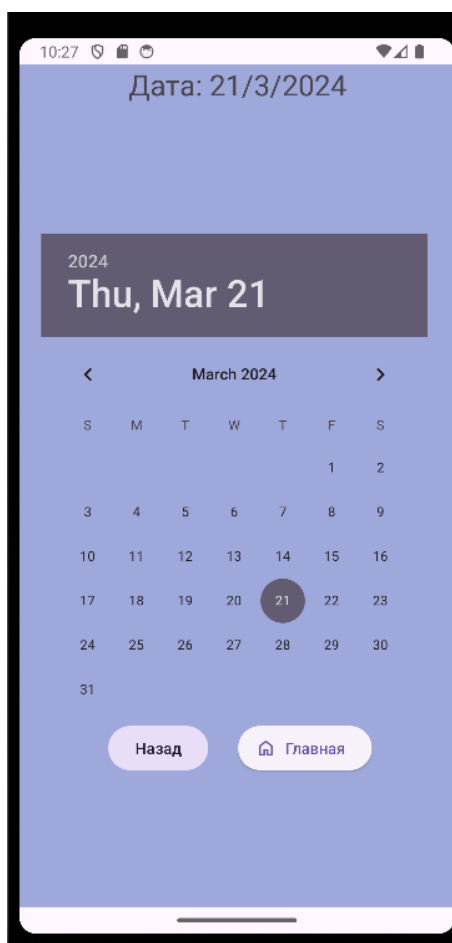


Рисунок 14 – тестирование кода для DatePicker



## 11. Элемент TimePicker Реализуйте пример.

Листинг 16 – реализация TimePicker.

```
<TextView android:id="@+id/timeTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="26sp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
<TimePicker android:id="@+id/timePicker"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/timeTextView" />
```

Листинг 17 – работа с TimePicker в коде на Java.

```
TextView timeTextView = binding.timeTextView;
TimePicker timePicker = binding.timePicker;
timePicker.setOnTimeChangedListener((view1, hourOfDay, minute) ->
    timeTextView.setText("Время: " + hourOfDay + ":" + minute));
```

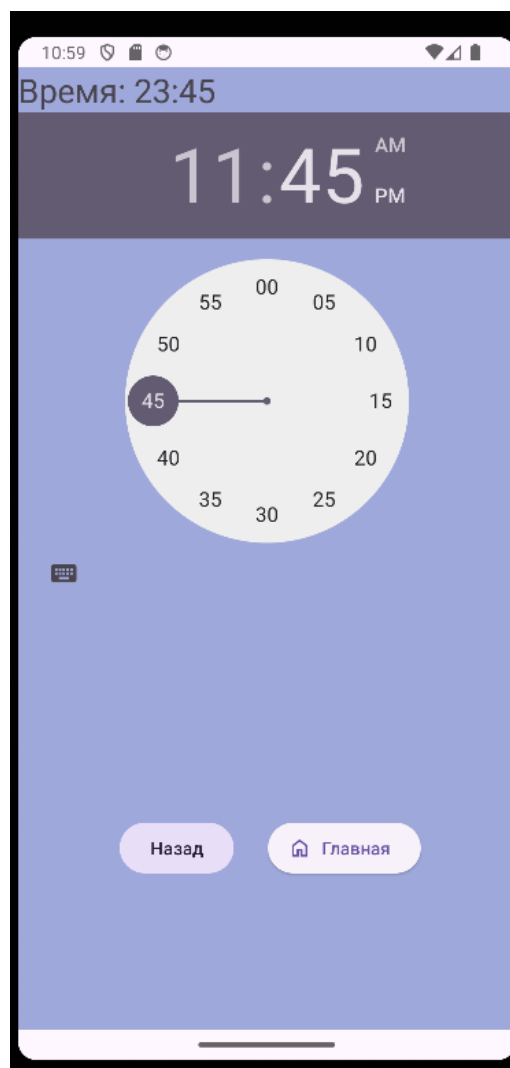


Рисунок 15 – тестирование кода для TimePicker

## 12. Элемент SeekBar (Ползунок). Реализуйте пример

*Листинг 18 – реализация SeekBar.*

```
<TextView
    android:id="@+id/seekBarValue"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:textSize="26sp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<SeekBar
    android:id="@+id/seekBar"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:max="100"
    android:progress="20"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    tools:layout_editor_absoluteY="228dp" />
```

*Листинг 19 – работа с SeekBar в коде на Java.*

```
SeekBar seekBar = binding.seekBar;
TextView textView = binding.seekBarValue;
seekBar.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SearchBar seekBar, int progress,
boolean fromUser) {
        textView.setText(String.valueOf(progress));
    }
    @Override
    public void onStartTrackingTouch(SearchBar seekBar) {
    }
    @Override
    public void onStopTrackingTouch(SearchBar seekBar) {
    }
});
```



Рисунок 16 – тестирование кода для SeekBar

## **Вывод**

В ходе работы мне удалось реализовать различные элементы управления (TextView, Toast, RadioButton, ToggleButton Snackbar, Checkbox, DatePicker, TimePicker, SeekBar в коде на Java и xml.