|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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 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. 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); |

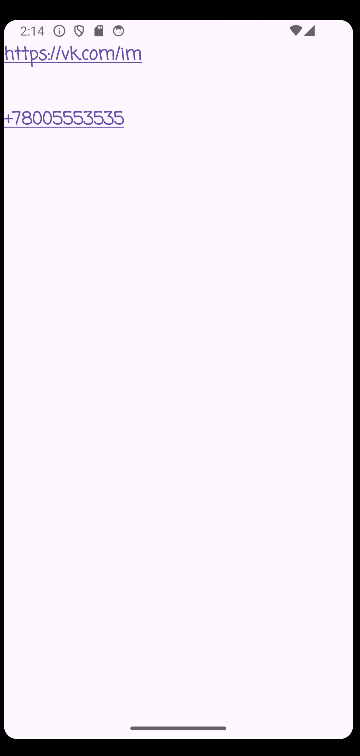


Рисунок 1 – тестирование кода для TextView с почтой и номером телефона

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); |

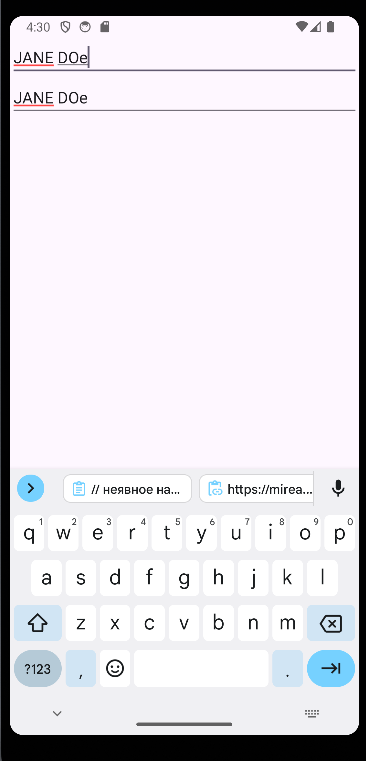


Рисунок 2 – тестирование кода для EditText

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" /> |

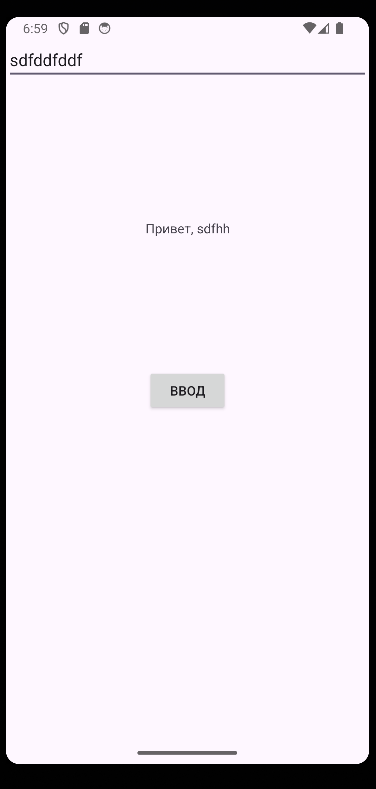


Рисунок 3 – тестирование кода для Button

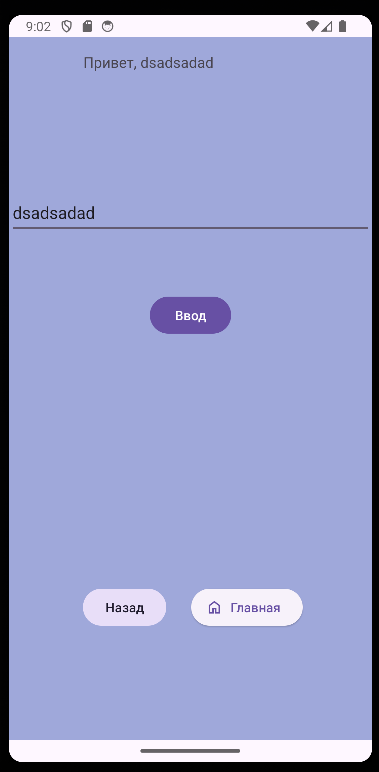


Рисунок 4 – тестирование xml для Button

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 - колическо 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(); }); |

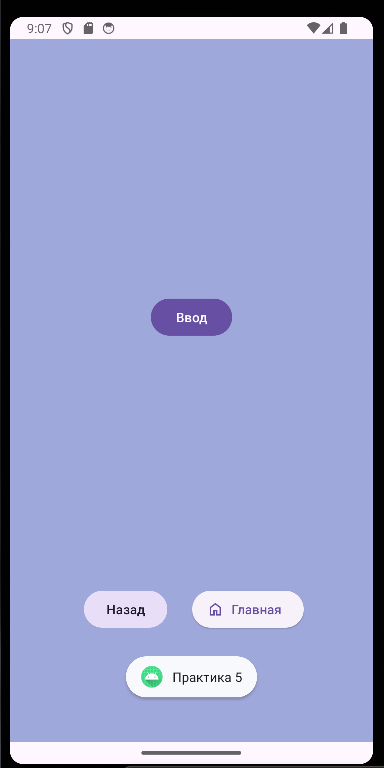


Рисунок 5 – тестирование кода для Toast с необходимыми параметрами

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(); }); |

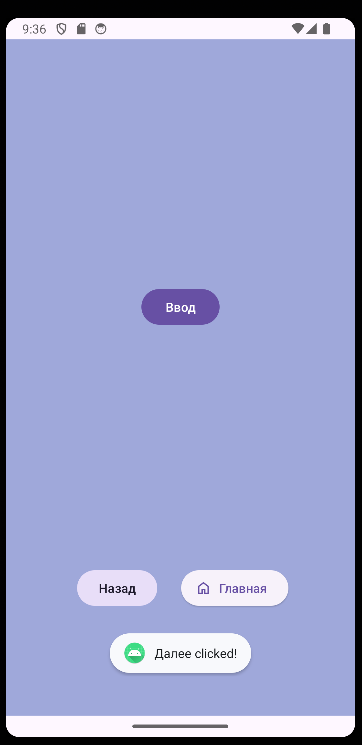
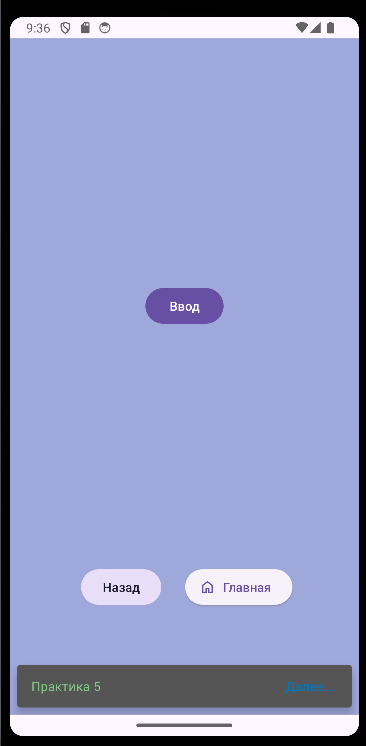


Рисунок 6,7 – тестирование кода для Snackbar

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();  } |

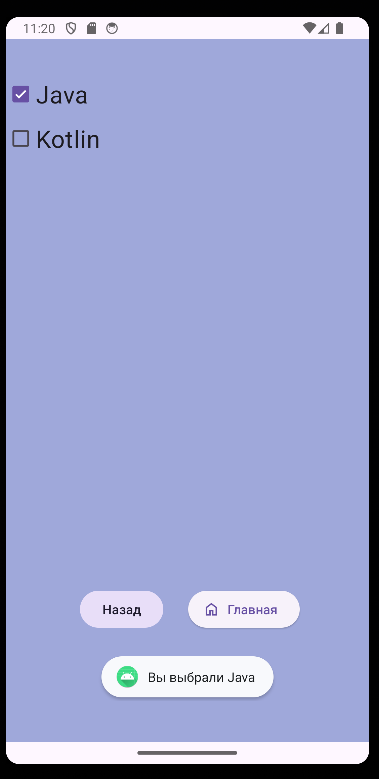


Рисунок 8 – тестирование кода для Checkbox

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");  } }); |

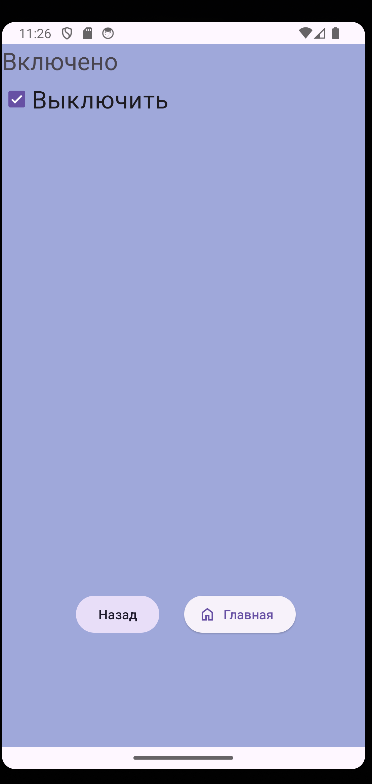


Рисунок 9 – тестирование кода для OnCheckedChangeListener

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); |

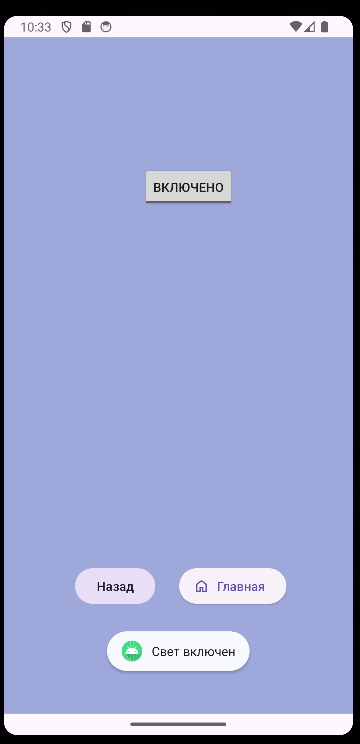
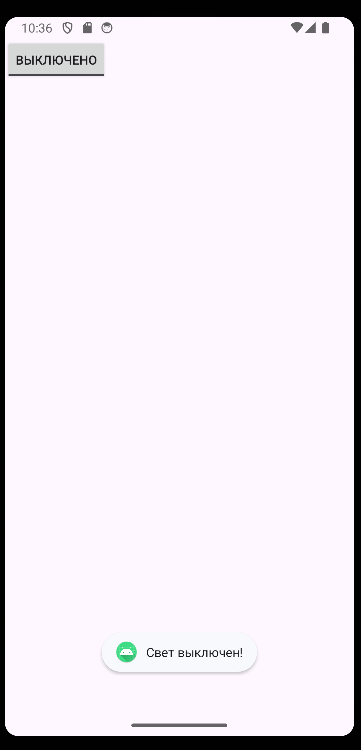
 

Рисунок 10,11 – тестирование кода для ToggleButton

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");  }  } } |

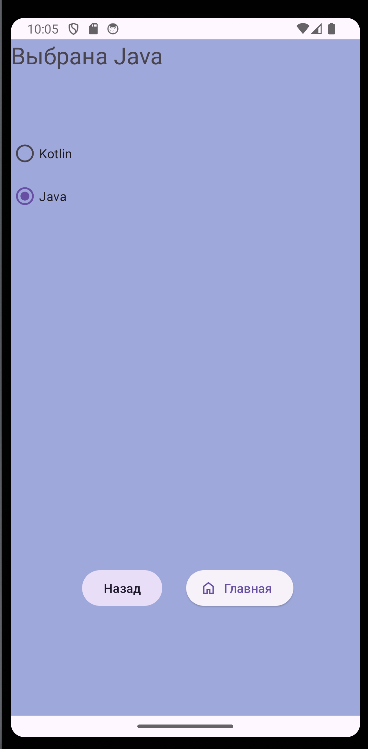
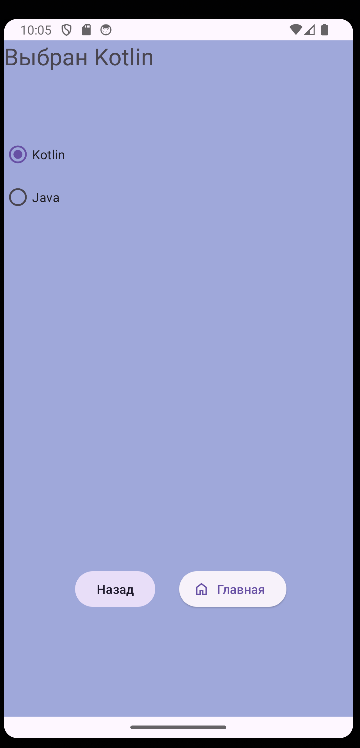
 

Рисунок 12,13 – тестирование кода для RadioButton

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())); |

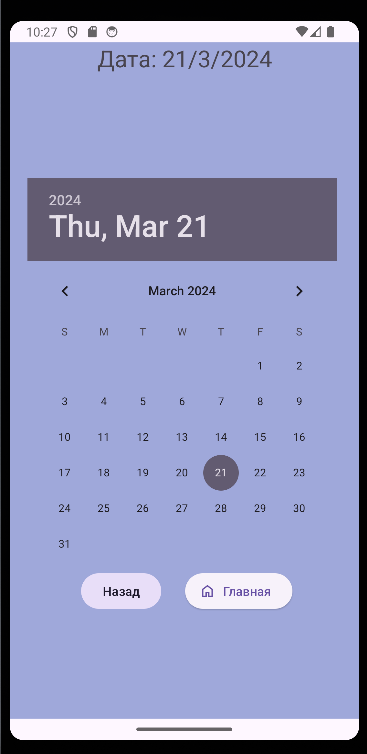


Рисунок 14 – тестирование кода для DatePicker

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)); |

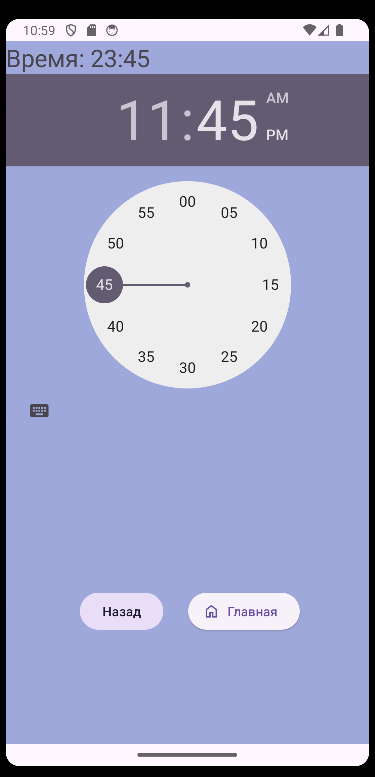


Рисунок 15 – тестирование кода для TimePicker

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(SeekBar seekBar, int progress, boolean fromUser) {  textView.setText(String.valueOf(progress));  }  @Override  public void onStartTrackingTouch(SeekBar seekBar) {  }  @Override  public void onStopTrackingTouch(SeekBar seekBar) {  } }); |

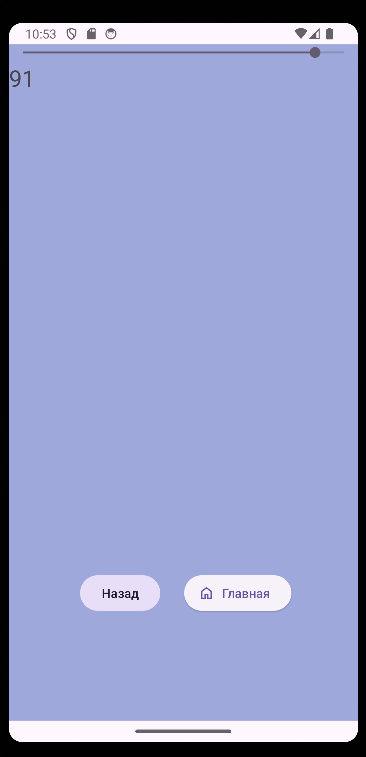


Рисунок 16 – тестирование кода для SeekBar

# 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.