Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

Лабораторная работа №2

по дисциплине "Технологии разработки мобильных приложений"

ТЕМА РАБОТЫ:

Изучение компонентов пользовательского интерфейса и обработчиков событий

Выполнил:

студент гр. ПРИ-120

Парахин К.В.

Принял:

преподаватель кафедры ИСПИ

Рощина А.И.

Владимир 2023 г

Цель работы:

Изучить основные графические компоненты мобильного приложения и механизмы организации обработки событий.

Выполнение работы:

Создадим шаблонный проект Android-приложения с пустой активностью.

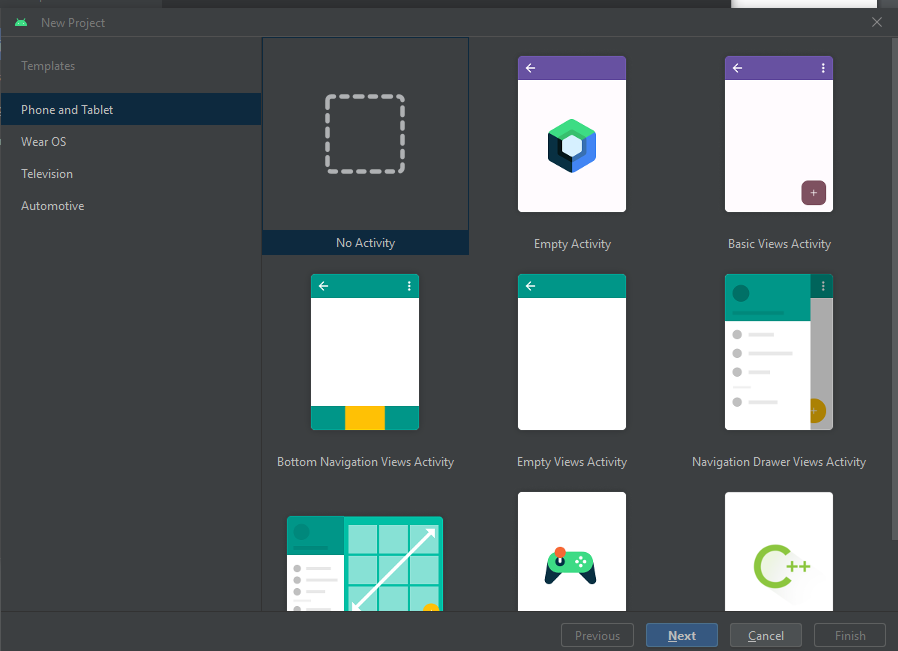


Рисунок 1. Создание пустого проекта

При создании указываем основные настройки проекта:

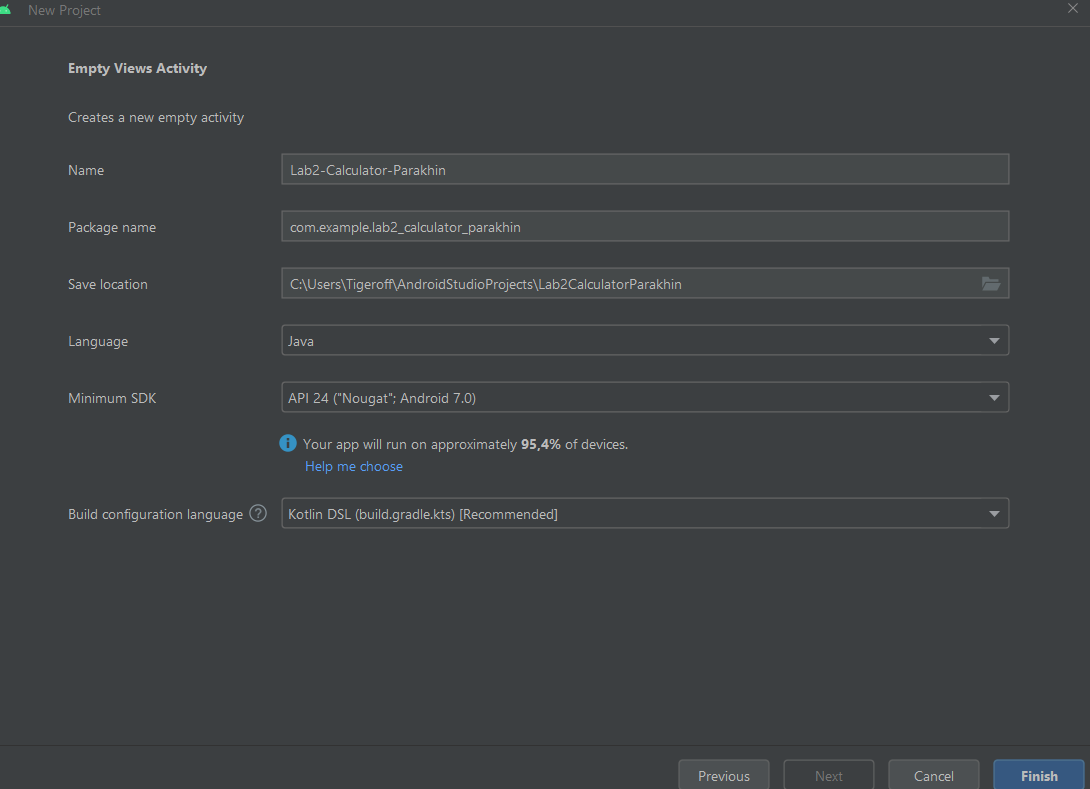


Рисунок 2. Настройка проекта

Далее перейдем в режим графического представления файла activity\_main.xml.

Добавляем туда поле для ввода-вывода текста и основные кнопки для ввода цифр и для выполнения базовых арифметических операций.

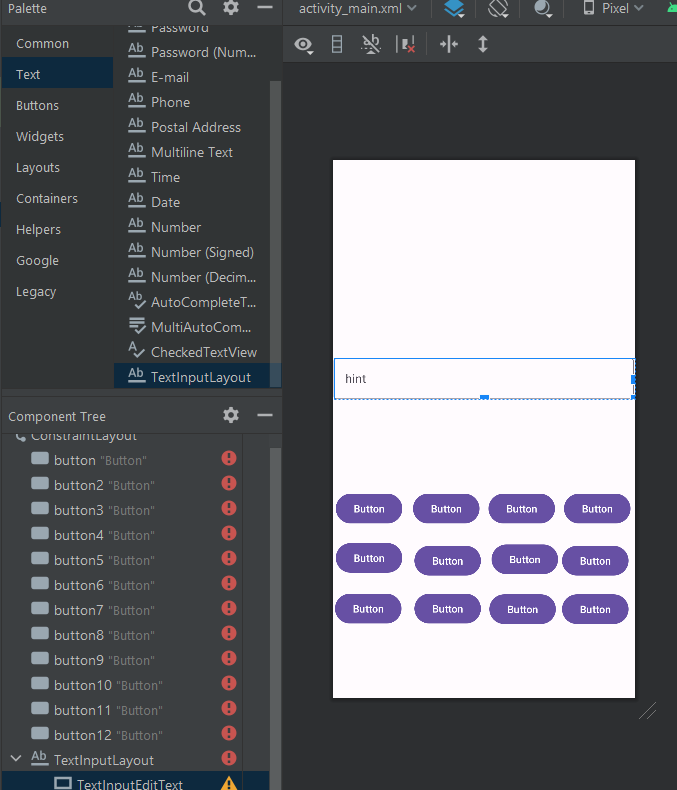


Рисунок 3. Добавление графических элементов из панели Palette

Необходим настроить вид, содержание и функциональную отзывчивость добавленных графических элементов управления.

Для этого перейдем в режим кода в том же файле activity\_main.xml и изменим содержимое внутри тегов:

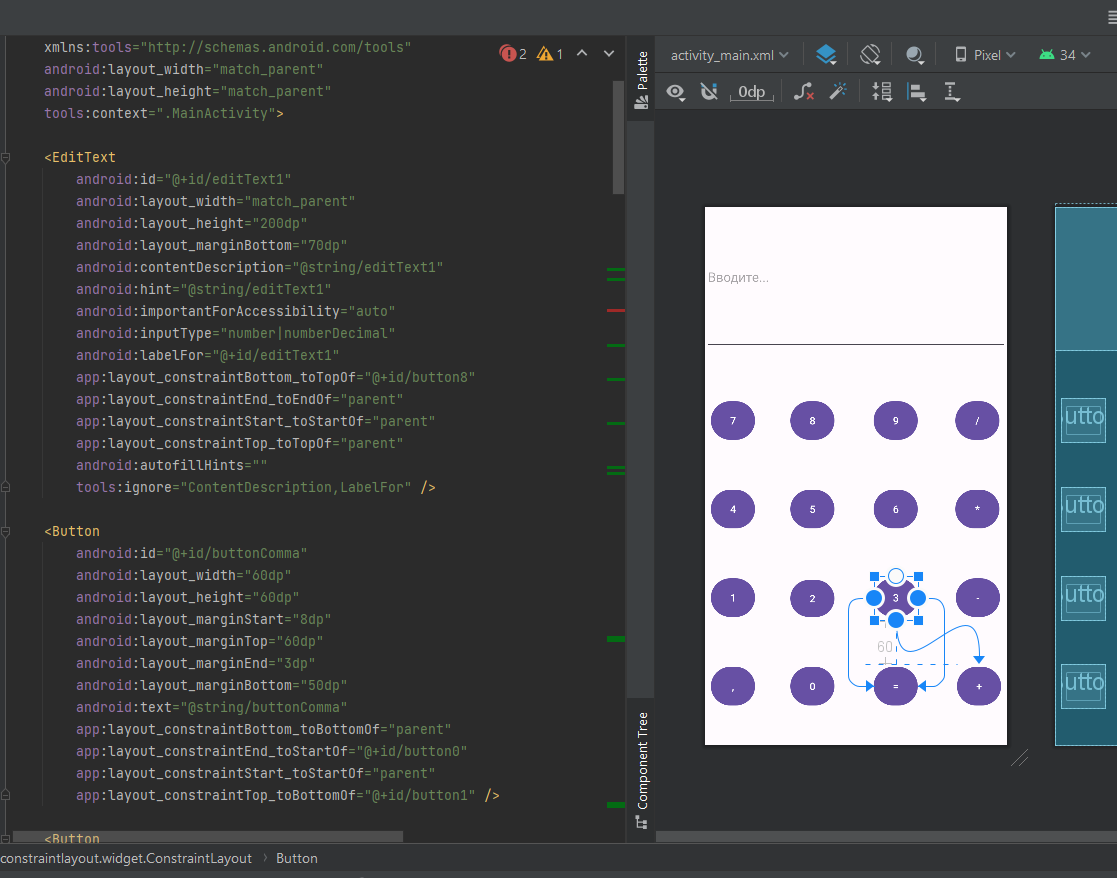


Рисунок 4. Размещение элементов на форме

Названия кнопок записываем как теги в отдельный файл strings.xml:

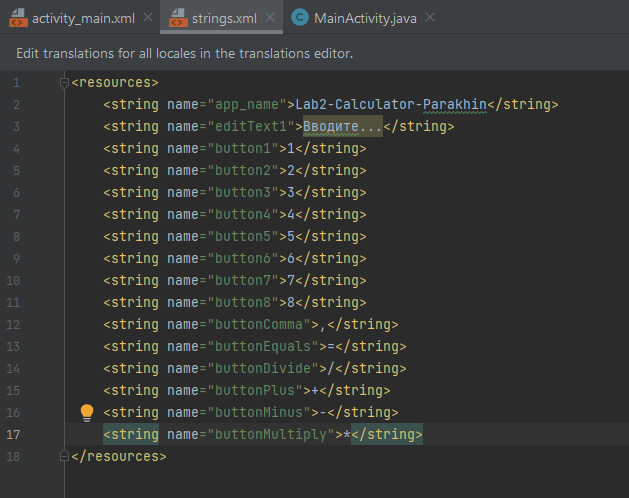


Рисунок 5. Файл strings.xml

Теперь перейдем в файл MainActivity.java. В нем создадим обработчики для нажатий на кнопки цифр и напишем логику для всех базовых операций (сложение, вычитание, умножение и деление).

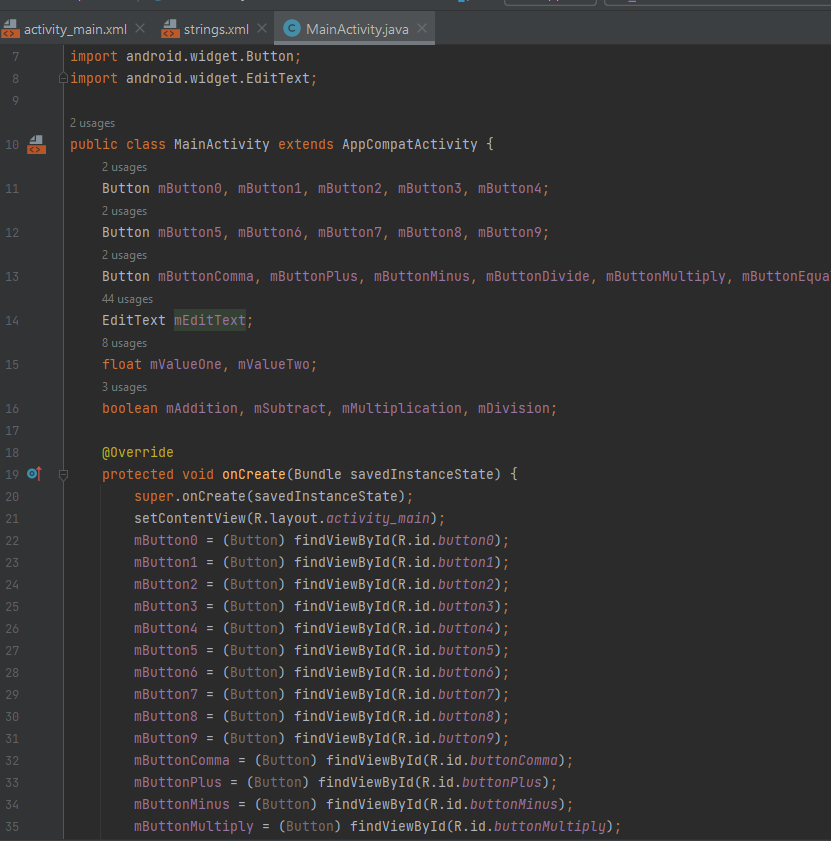


Рисунок 6. Код калькулятора

Индивидуальное задание:

Вариант 1:

1) Изменить логику таким образом, чтобы было возможно производить операции над двумя и больше числами;

2) Добавить операции взятия корня числа и возведения в квадрат;

3) Добавить возможность сохранять промежуточный результат и очищать его путем реализации кнопок M и MC.

Код MainActivity.java:

public class MainActivity extends AppCompatActivity {  
 Button mButton0, mButton1, mButton2, mButton3, mButton4;  
 Button mButton5, mButton6, mButton7, mButton8, mButton9;  
 Button mButtonComma, mButtonPlus, mButtonMinus, mButtonDivide, mButtonMultiply;  
 Button mButtonEquals, mButtonSqr, mButtonSqrt, mButtonMS, mButtonM;  
 EditText mEditText;  
 String [] Mvalues = new String[15];  
 int currentIndex;  
 String wholeText;  
 String currentText;  
 boolean mAddition, mSubtract, mMultiplication, mDivision, mSqr, mSqrt;  
  
 private String provideOperations(){  
  
 float commonResult = 0;  
  
 int vipOpsCount = CalculateVipOpsCount();  
  
 String[] newValues = Mvalues;  
  
 for(int i = 0; i <= vipOpsCount; i++){  
 newValues = makeOneLessVipOperation(newValues);  
 }  
  
 commonResult = provideBaseOperations(newValues);  
  
 return Float.*toString*(commonResult);  
 }  
  
 private int CalculateVipOpsCount(){  
 int n = 0;  
 for(int i = 0; i < Mvalues.length - 1; i++){  
 if (Mvalues[i].equals("\*") || Mvalues[i].equals("/")  
 || Mvalues[i].equals("sqr") || Mvalues[i].equals("sqrt")){  
 n = n + 1;  
 }  
 }  
 return n;  
 }  
  
 private String[] makeOneLessVipOperation(String[] values){  
 String[] newValues = new String[15];  
  
 for(int i = 0; i < newValues.length; i++){  
 newValues[i] = "";  
 }  
  
 int k = 0;  
  
 int sqrCount = 0;  
  
 boolean vipOperationPerformed = false;  
  
 for(int j = 0; j < values.length - 1; j = j + 1) {  
  
 if (values[k].equals("\*") && !vipOperationPerformed) {  
 float curResult = Float.*parseFloat*(values[k - 1]) \* Float.*parseFloat*(values[k + 1]);  
 newValues[j - 1] = Float.*toString*(curResult);  
 vipOperationPerformed = true;  
 k += 2;  
 } else if (values[k].equals("/") && !vipOperationPerformed) {

if (Float.*parseFloat*(values[k + 1]) == 0){  
 throw new IllegalArgumentException("Divide by zero");  
}  
 float curResult = Float.*parseFloat*(values[k - 1]) / Float.*parseFloat*(values[k + 1]);  
 newValues[j - 1] = Float.*toString*(curResult);  
 vipOperationPerformed = true;  
 k += 2;  
 } else if (values[k].equals("sqr") && !vipOperationPerformed) {  
 float curResult = Float.*parseFloat*(values[k + 1]) \* Float.*parseFloat*(values[k + 1]);  
 newValues[j] = Float.*toString*(curResult);  
 vipOperationPerformed = true;  
 k += 2;  
 sqrCount = sqrCount == 0 ? 1 : 0;  
 } else if (values[k].equals("sqrt") && !vipOperationPerformed) {  
 double curResult = Math.*sqrt*(Float.*parseFloat*(values[k + 1]));  
 newValues[j] = Double.*toString*(curResult);  
 vipOperationPerformed = true;  
 k += 2;  
 sqrCount = sqrCount == 0 ? 1 : 0;  
 } else {  
 int d = k - j - sqrCount;  
 newValues[j - d] = values[k];  
 k += 1;  
 }  
 }  
  
 return newValues;  
 }  
  
 private float provideBaseOperations(String[] newValues){  
 float result = 0;  
 int i = 0;  
 result = Float.*parseFloat*(newValues[i]);  
 while (!newValues[i + 1].equals("")){  
 if (newValues[i + 1].equals("+")){  
 result += Float.*parseFloat*(newValues[i + 2]);  
 }  
 if (newValues[i + 1].equals("-")){  
 result -= Float.*parseFloat*(newValues[i + 2]);  
 }  
 i += 2;  
 }  
 return result;  
 }  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 mButton0 = (Button) findViewById(R.id.*button0*);  
 mButton1 = (Button) findViewById(R.id.*button1*);  
 mButton2 = (Button) findViewById(R.id.*button2*);  
 mButton3 = (Button) findViewById(R.id.*button3*);  
 mButton4 = (Button) findViewById(R.id.*button4*);  
 mButton5 = (Button) findViewById(R.id.*button5*);  
 mButton6 = (Button) findViewById(R.id.*button6*);  
 mButton7 = (Button) findViewById(R.id.*button7*);  
 mButton8 = (Button) findViewById(R.id.*button8*);  
 mButton9 = (Button) findViewById(R.id.*button9*);  
 mButtonComma = (Button) findViewById(R.id.*buttonComma*);  
 mButtonPlus = (Button) findViewById(R.id.*buttonPlus*);  
 mButtonMinus = (Button) findViewById(R.id.*buttonMinus*);  
 mButtonMultiply = (Button) findViewById(R.id.*buttonMultiply*);  
 mButtonDivide = (Button) findViewById(R.id.*buttonDivide*);  
 mButtonEquals = (Button) findViewById(R.id.*buttonEquals*);  
 mButtonSqr = (Button) findViewById(R.id.*buttonSqr*);  
 mButtonSqrt = (Button) findViewById(R.id.*buttonSqrt*);  
 mButtonMS = (Button) findViewById(R.id.*buttonMS*);  
 mButtonM = (Button) findViewById(R.id.*buttonM*);  
  
 mEditText = (EditText) findViewById(R.id.*editText1*);  
  
 for(int i = 0; i < Mvalues.length; i++){  
 Mvalues[i] = "";  
 }  
  
 currentText = "";  
 wholeText = "";  
  
 currentIndex = 0;  
  
 mButton1.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (currentText.equals("0")){  
 return;  
 }  
 currentText += "1";  
 wholeText += "1";  
 mEditText.setText(wholeText);  
 }  
 });  
  
 mButton2.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (currentText.equals("0")){  
 return;  
 }  
 currentText += "2";  
 wholeText += "2";  
 mEditText.setText(wholeText);  
 }  
 });  
  
 mButton3.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (currentText.equals("0")){  
 return;  
 }  
 currentText += "3";  
 wholeText += "3";  
 mEditText.setText(wholeText);  
 }  
 });  
  
 mButton4.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (currentText.equals("0")){  
 return;  
 }  
 currentText += "4";  
 wholeText += "4";  
 mEditText.setText(wholeText);  
 }  
 });  
 mButton5.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (currentText.equals("0")){  
 return;  
 }  
 currentText += "5";  
 wholeText += "5";  
 mEditText.setText(wholeText);  
 }  
 });  
  
 mButton6.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (currentText.equals("0")){  
 return;  
 }  
 currentText += "6";  
 wholeText += "6";  
 mEditText.setText(wholeText);  
 }  
 });  
  
 mButton7.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (currentText.equals("0")){  
 return;  
 }  
 currentText += "7";  
 wholeText += "7";  
 mEditText.setText(wholeText);  
 }  
 });  
  
 mButton8.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (currentText.equals("0")){  
 return;  
 }  
 currentText += "8";  
 wholeText += "8";  
 mEditText.setText(wholeText);  
 }  
 });  
  
 mButton9.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (currentText.equals("0")){  
 return;  
 }  
 currentText += "9";  
 wholeText += "9";  
 mEditText.setText(wholeText);  
 }  
 });  
  
 mButton0.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (currentText.equals("0")){  
 return;  
 }  
 currentText += "0";  
 wholeText += "0";  
 mEditText.setText(wholeText);  
 }  
 });  
  
 mButtonPlus.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (mEditText == null) {  
 mEditText.setText("");  
 } else {  
 int index = currentIndex;  
 if (index >= Mvalues.length - 1){  
 return;  
 }  
 if (currentText.equals("")){  
 return;  
 }  
 if (currentText.endsWith(".")){  
 return;  
 }  
 Mvalues[index] = currentText;  
 currentText = "";  
 wholeText += "+";  
 mEditText.setText(wholeText);  
 Mvalues[index + 1] = "+";  
 currentIndex += 2;  
 mAddition = true;  
 }  
 }  
 });  
  
 mButtonMinus.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (mEditText == null) {  
 mEditText.setText("");  
 } else {  
 int index = currentIndex;  
 if (index >= Mvalues.length - 1){  
 return;  
 }  
 if (currentText.equals("")){  
 return;  
 }  
 if (currentText.endsWith(".")){  
 return;  
 }  
 Mvalues[index] = currentText;  
 currentText = "";  
 wholeText += "-";  
 mEditText.setText(wholeText);  
 Mvalues[index + 1] = "-";  
 currentIndex += 2;  
 mSubtract = true;  
 }  
 }  
 });  
  
 mButtonMultiply.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (mEditText == null) {  
 mEditText.setText("");  
 } else {  
 int index = currentIndex;  
 if (index >= Mvalues.length - 1){  
 return;  
 }  
 if (currentText.equals("")){  
 return;  
 }  
 if (currentText.endsWith(".")){  
 return;  
 }  
 Mvalues[index] = currentText;  
 currentText = "";  
 wholeText += "\*";  
 mEditText.setText(wholeText);  
 Mvalues[index + 1] = "\*";  
 currentIndex += 2;  
 mMultiplication = true;  
 }  
 }  
 });  
  
 mButtonDivide.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (mEditText == null) {  
 mEditText.setText("");  
 } else {  
 int index = currentIndex;  
 if (index >= Mvalues.length - 1){  
 return;  
 }  
 if (currentText.equals("")){  
 return;  
 }  
 if (currentText.endsWith(".")){  
 return;  
 }  
 Mvalues[index] = currentText;  
 currentText = "";  
 wholeText += "/";  
 mEditText.setText(wholeText);  
 Mvalues[index + 1] = "/";  
 currentIndex += 2;  
 mDivision = true;  
 }  
 }  
 });  
  
 mButtonEquals.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (currentText.equals("")){  
 return;  
 }  
 if (wholeText.equals("")){  
 return;  
 }  
  
 if (Mvalues[currentIndex].equals("sqr")  
 || Mvalues[currentIndex].equals("sqrt")){  
 Mvalues[currentIndex + 1] = currentText;  
 }  
 else{  
 Mvalues[currentIndex] = currentText;  
 }  
  
 String result = provideOperations();  
  
 currentIndex = 0;  
  
 for(int i = 0; i < Mvalues.length; i++){  
 Mvalues[i] = "";  
 }  
  
 currentText = result;  
 wholeText = result;  
  
 Mvalues[currentIndex] = currentText;  
  
 mEditText.setText(wholeText);  
 }  
 });  
  
 mButtonComma.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (currentText.equals(".") || currentText.equals("")){  
 return;  
 }  
 currentText += ".";  
 wholeText += ".";  
 mEditText.setText(wholeText);  
 }  
 });  
  
 mButtonSqr.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (mEditText == null) {  
 mEditText.setText("");  
 } else {  
 int index = currentIndex;  
 if (index >= Mvalues.length - 1){  
 return;  
 }  
 if (!currentText.equals("")){  
 return;  
 }  
 Mvalues[currentIndex] = "sqr";  
 wholeText += "sqr";  
 mEditText.setText(wholeText);  
 mSqr = true;  
 }  
 }  
 });  
  
 mButtonSqrt.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (mEditText == null) {  
 mEditText.setText("");  
 } else {  
 int index = currentIndex;  
 if (index >= Mvalues.length - 1){  
 return;  
 }  
 if (!currentText.equals("")){  
 return;  
 }  
 Mvalues[currentIndex] = "sqrt";  
 wholeText += "sqrt";  
 mEditText.setText(wholeText);  
 mSqrt = true;  
 }  
 }  
 });  
  
 mButtonMS.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (mEditText == null) {  
 mEditText.setText("");  
 } else {  
 currentText = "";  
 currentIndex = 0;  
 mEditText.setText(currentText);  
 }  
 }  
 });  
  
 mButtonM.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (mEditText == null) {  
 mEditText.setText("");  
 } else {  
 currentText = "";  
 wholeText = "";  
 currentIndex = 0;  
 mEditText.setText(wholeText);  
 }  
 }  
 });  
 }

Для этого были добавлены 4 новые кнопки: ^2 (sqr), sqrt, MS (очищение поля ввода с сохранением в переменную), M – полное очищение промежуточного результата и поля ввода.

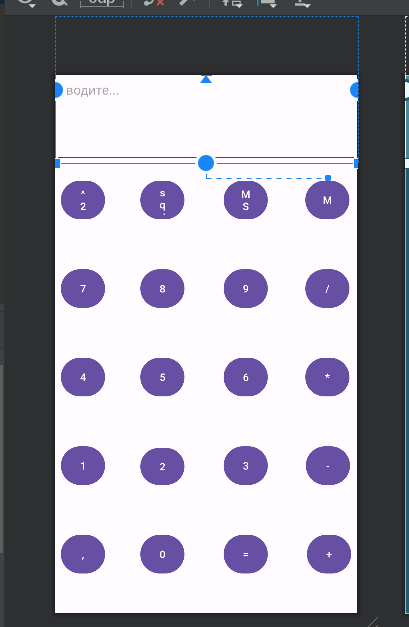


Рисунок 7. Панель активности

Тестирование калькулятора в эмуляторе:

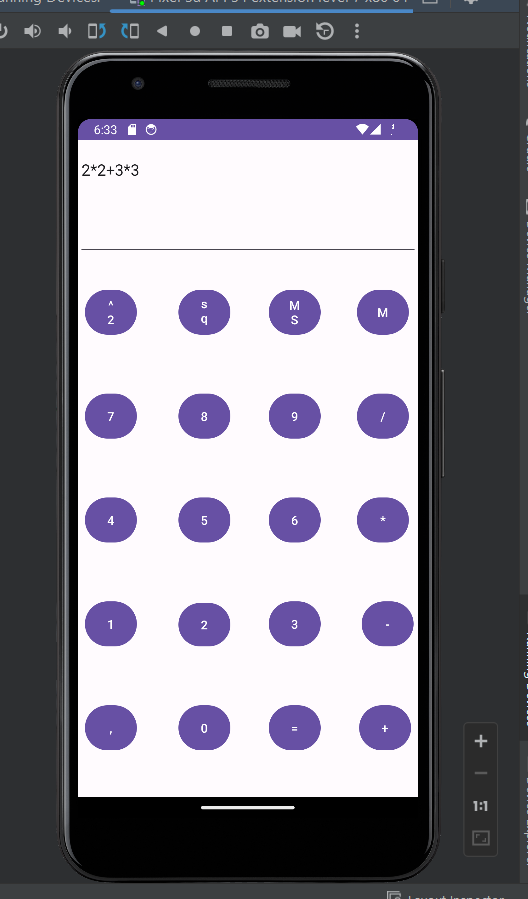


Рисунок 1. Ввод в поле нескольких операций

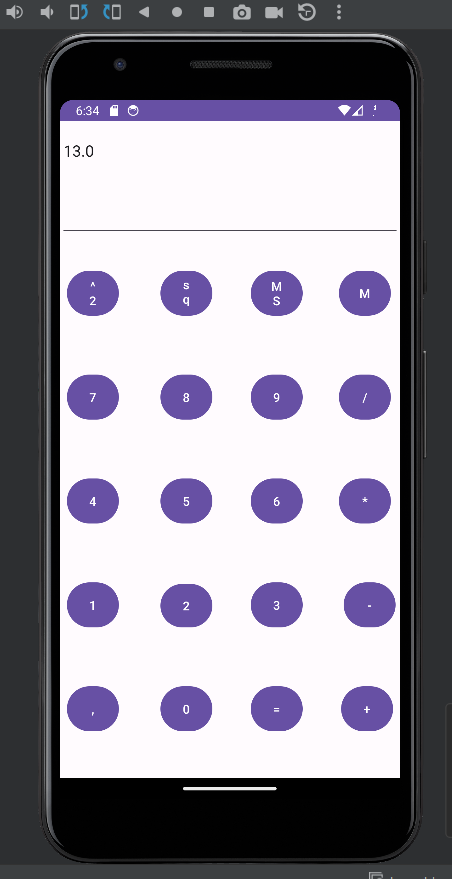


Рисунок 2. Получение корректного результата

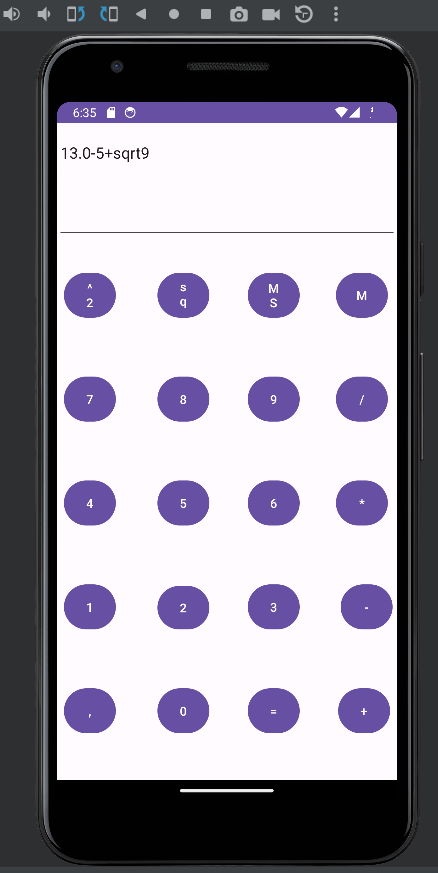


Рисунок 3. Ввод выражения, используя старый результат (добавляя операцию взятия корня в конце)

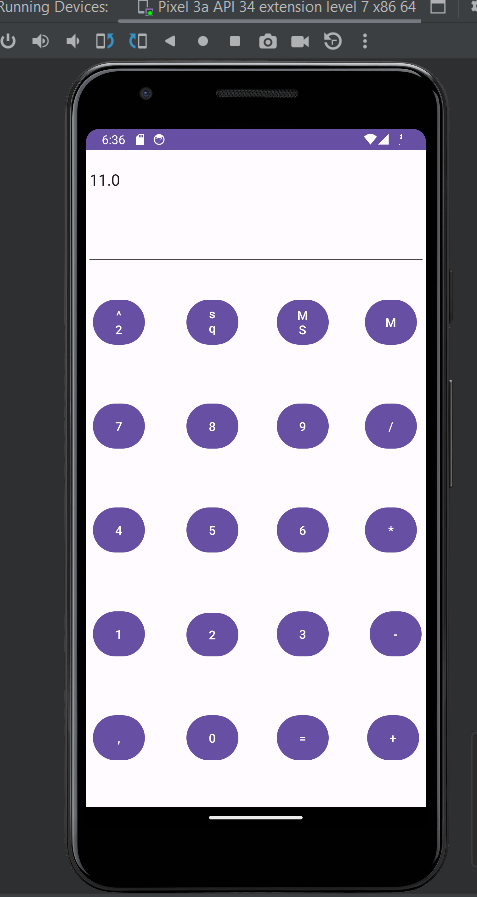


Рисунок 4. Полученный корректный результат

По аналогии работают все остальные операции в разных комбинациях, деление на ноль приводит при этом к выбрасыванию исключения.

Вывод

В результате выполнения работы, я изучил основные графические компоненты мобильного приложения и механизмы организации обработки событий.