**Отчёт по практической работе «Лабораторная работа 14-15»**

Выполнил студент

Курс: ПРИ-311

ФИО: Пикалов Е.А.

**Задание:**

Модифицируйте класс Calculator.java так, чтобы все кнопки с цифрами хранились в массиве с десятью элементами, который должен быть объявлен вот так: Buttons[] numButtons= new Buttons[10]; Замените 10 строк кода, которые начинаются с button0=new JButton("0"); циклом, который создаёт кнопки и добавляет их в массив. Подсказка: загляните в исходный код игры Крестики-­­Нолики в главе\_\_

Практические упражнения Попробуйте разделить число на ноль с помощью нашего калькулятора -­­ дисплей покажет слово Infinity. Измените класс CalculatorEngine, чтобы отображалось сообщение ”На ноль делить нельзя”, если пользователь нажмет на кнопку ”Разделить”, когда дисплей калькулятора будет пуст.

Измените класс CalculatorEngine , чтобы запретить вводить больше одной точки в числе. Подсказка: прочтите про метод indexOf() класса String, чтобы узнать, есть ли уже в числе точка.\_\_

**Программный код:**

**App.java**

public class App {

    public static void main(String[] args) throws Exception {

        Calculator calculator = new Calculator();

        calculator.setVisible(true);

    }

}

**Calculator.java**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class Calculator extends JFrame {

    private char sign;

    private String text;

    private JButton[] buttons;

    private String save\_text;

    private Container container;

    private JPanel panel1;

    private JPanel panel2;

    private JTextField textfield;

    private JButton button0;

    private JButton button1;

    private JButton button2;

    private JButton button3;

    private JButton button4;

    private JButton button5;

    private JButton button6;

    private JButton button7;

    private JButton button8;

    private JButton button9;

    private JButton plus;

    private JButton minus;

    private JButton umnozit;

    private JButton delit;

    private JButton ravno;

    private JButton delete;

    private JButton point;

    private JButton clean;

    private JButton download;

    private JButton onload;

    public Calculator(){

        super("Калькулятор");

        super.setBounds(200, 200, 220, 270);

        super.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        super.setResizable(false);

        container = super.getContentPane();

        container.setLayout(new BorderLayout());

        sign = ' ';

        buttons = new JButton[4];

        save\_text = "0";

        panel1 = new JPanel();

        panel2 = new JPanel();

        textfield = new JTextField(20);

        button1 = new JButton("1");

        button2 = new JButton("2");

        button3 = new JButton("3");

        button4 = new JButton("4");

        button5 = new JButton("5");

        button6 = new JButton("6");

        button7 = new JButton("7");

        button8 = new JButton("8");

        button9 = new JButton("9");

        button0 = new JButton("0");

        plus = new JButton("+");

        minus = new JButton("-");

        umnozit = new JButton("\*");

        delit = new JButton("/");

        ravno = new JButton("=");

        clean = new JButton("c");

        point = new JButton(",");

        delete = new JButton("<");

        download = new JButton("s");

        onload = new JButton("g");

        buttons[0] = plus;

        buttons[1] = minus;

        buttons[2] = umnozit;

        buttons[3] = delit;

        button1.addActionListener(new ButtonEvent());

        button1.setFocusPainted(false);

        button1.setBackground(Color.BLACK);

        button1.setForeground(Color.WHITE);

        button1.setBorder(null);

        button2.addActionListener(new ButtonEvent());

        button2.setFocusPainted(false);

        button2.setBackground(Color.BLACK);

        button2.setForeground(Color.WHITE);

        button2.setBorder(null);

        button3.addActionListener(new ButtonEvent());

        button3.setFocusPainted(false);

        button3.setBackground(Color.BLACK);

        button3.setForeground(Color.WHITE);

        button3.setBorder(null);

        button4.addActionListener(new ButtonEvent());

        button4.setFocusPainted(false);

        button4.setBackground(Color.BLACK);

        button4.setForeground(Color.WHITE);

        button4.setBorder(null);

        button5.addActionListener(new ButtonEvent());

        button5.setFocusPainted(false);

        button5.setBackground(Color.BLACK);

        button5.setForeground(Color.WHITE);

        button5.setBorder(null);

        button6.addActionListener(new ButtonEvent());

        button6.setFocusPainted(false);

        button6.setBackground(Color.BLACK);

        button6.setForeground(Color.WHITE);

        button6.setBorder(null);

        button7.addActionListener(new ButtonEvent());

        button7.setFocusPainted(false);

        button7.setBackground(Color.BLACK);

        button7.setForeground(Color.WHITE);

        button7.setBorder(null);

        button8.addActionListener(new ButtonEvent());

        button8.setFocusPainted(false);

        button8.setBackground(Color.BLACK);

        button8.setForeground(Color.WHITE);

        button8.setBorder(null);

        button9.addActionListener(new ButtonEvent());

        button9.setFocusPainted(false);

        button9.setBackground(Color.BLACK);

        button9.setForeground(Color.WHITE);

        button9.setBorder(null);

        button0.addActionListener(new ButtonEvent());

        button0.setFocusPainted(false);

        button0.setBackground(Color.BLACK);

        button0.setForeground(Color.WHITE);

        button0.setBorder(null);

        plus.addActionListener(new SignEvent());

        plus.setFocusPainted(false);

        minus.addActionListener(new SignEvent());

        minus.setFocusPainted(false);

        umnozit.addActionListener(new SignEvent());

        umnozit.setFocusPainted(false);

        delit.addActionListener(new SignEvent());

        delit.setFocusPainted(false);

        ravno.addActionListener(new EquallyEvent());

        ravno.setFocusPainted(false);

        clean.addActionListener(new CleanEvent());

        clean.setFocusPainted(false);

        point.addActionListener(new PointEvent());

        point.setFocusPainted(false);

        delete.addActionListener(new DeleteEvent());

        delete.setFocusPainted(false);

        download.addActionListener(new DownloadEvent());

        download.setFocusPainted(false);

        onload.addActionListener(new DownloadEvent());

        onload.setFocusPainted(false);

        textfield.setPreferredSize(new Dimension(MAXIMIZED\_HORIZ, 30));

        textfield.setEditable(false);

        textfield.setHorizontalAlignment(JTextField.RIGHT);

        textfield.setText("0");

        panel1.setLayout(new GridLayout());

        panel1.setBorder(BorderFactory.createEmptyBorder(5, 5, 5, 5));

        panel2.setLayout(new GridLayout(5, 4, 10, 10));

        panel2.setBorder(BorderFactory.createEmptyBorder(5, 5, 5, 5));

        panel1.add(textfield);

        panel2.add(button1);

        panel2.add(button2);

        panel2.add(button3);

        panel2.add(plus);

        panel2.add(button4);

        panel2.add(button5);

        panel2.add(button6);

        panel2.add(minus);

        panel2.add(button7);

        panel2.add(button8);

        panel2.add(button9);

        panel2.add(umnozit);

        panel2.add(point);

        panel2.add(button0);

        panel2.add(ravno);

        panel2.add(delit);

        panel2.add(clean);

        panel2.add(delete);

        panel2.add(download);

        panel2.add(onload);

        container.add(panel1, BorderLayout.NORTH);

        container.add(panel2, BorderLayout.CENTER);

    }

    class ButtonEvent implements ActionListener{

        @Override

        public void actionPerformed(ActionEvent e) {

            if (textfield.getText().equals("На ноль делить нельзя")){

                textfield.setText(e.getActionCommand());

            }

            else{

                if (textfield.getText().equals("0")){

                    textfield.setText(e.getActionCommand());

                }

                else{

                    textfield.setText(textfield.getText() + e.getActionCommand());

                }

            }

        }

    }

    class SignEvent implements ActionListener{

        @Override

        public void actionPerformed(ActionEvent e) {

            if (!textfield.getText().equals("На ноль делить нельзя")){

                sign = e.getActionCommand().charAt(0);

                for (JButton but : buttons){

                    for (ActionListener al : but.getActionListeners()){

                        but.removeActionListener(al);

                    }

                }

                textfield.setText(textfield.getText() + e.getActionCommand());

            }

        }

    }

    class EquallyEvent implements ActionListener{

        @Override

        public void actionPerformed(ActionEvent e) {

            text = textfield.getText();

            String[] value;

            String result;

            boolean if\_minus;

            boolean if\_sign = true;

            boolean if\_nol = false;

            boolean if\_text = false;

            if (text.equals("На ноль делить нельзя")){

                if\_text = true;

            }

            if (text.startsWith("-") && sign == '-'){

                if\_minus = true;

                text = text.substring(1, text.length());

            }

            else{

                if\_minus = false;

            }

            value = text.split("\\" + sign);

            if (if\_text != true){

                if (!text.endsWith("-") && !text.endsWith("+") && !text.endsWith("/") && !text.endsWith("\*")){

                    switch (sign) {

                        case '+':

                            if\_sign = true;

                            result = Double.toString(Double.parseDouble(value[0]) + Double.parseDouble(value[1]));

                            System.out.println(value[0] + "+" + value[1] + "=" + result);

                            break;

                        case '-':

                            if\_sign = true;

                            if (if\_minus){

                                result = Double.toString(Double.parseDouble("-" + value[0]) - Double.parseDouble(value[1]));

                                System.out.println("-" + value[0] + "-" + value[1] + "=" + result);

                            }

                            else{

                                result = Double.toString(Double.parseDouble(value[0]) - Double.parseDouble(value[1]));

                                System.out.println(value[0] + "-" + value[1] + "=" + result);

                            }

                            break;

                        case '\*':

                            if\_sign = true;

                            result = Double.toString(Double.parseDouble(value[0]) \* Double.parseDouble(value[1]));

                            System.out.println(value[0] + "\*" + value[1] + "=" + result);

                            break;

                        case '/':

                            if (value[1].equals("0")){

                                result = "На ноль делить нельзя";

                                if\_sign = true;

                                if\_nol = true;

                                break;

                            }

                            else{

                                if\_sign = true;

                                result = Double.toString(Double.parseDouble(value[0]) / Double.parseDouble(value[1]));

                                System.out.println(value[0] + "/" + value[1] + "=" + result);

                                break;

                            }

                        default:

                            if\_sign = false;

                            result = Double.toString(Double.parseDouble(value[0]));

                            break;

                    }

                }

                else{

                    result = Double.toString(Double.parseDouble(value[0]));

                }

                if (!if\_nol){

                    if(result.split("\\.")[1].equals("0")){

                        result = result.substring(0, result.length()-2);

                    }

                }

            }

            else{

                result = "0";

            }

            textfield.setText(result);

            if (if\_text != true){

                if (if\_sign){

                    sign = ' ';

                    for (JButton but : buttons){

                        but.addActionListener(new SignEvent());

                    }

                }

            }

        }

    }

    class CleanEvent implements ActionListener{

        @Override

        public void actionPerformed(ActionEvent e) {

            textfield.setText("0");

            if (sign != ' '){

                sign = ' ';

                for (JButton but : buttons){

                    but.addActionListener(new SignEvent());

                }

            }

        }

    }

    class PointEvent implements ActionListener{

        @Override

        public void actionPerformed(ActionEvent e) {

            if (!textfield.getText().equals("На ноль делить нельзя")){

                String[] text = textfield.getText().split("(?<=[-+\*/])|(?=[-+\*/])");

                if (!text[text.length-1].equals("\*") && !text[text.length-1].equals("-") && !text[text.length-1].equals("+") && !text[text.length-1].equals("/") && text[text.length-1].indexOf(".") == -1){

                    textfield.setText(textfield.getText() + ".");

                }

            }

        }

    }

    class DeleteEvent implements ActionListener{

        @Override

        public void actionPerformed(ActionEvent e) {

            if (!textfield.getText().equals("На ноль делить нельзя")){

                if (sign != ' '){

                    sign = ' ';

                    for (JButton but : buttons){

                        but.addActionListener(new SignEvent());

                    }

                }

                if (textfield.getText().length() == 1){

                    textfield.setText("0");

                }

                else{

                    textfield.setText(textfield.getText().substring(0, textfield.getText().length()-1));

                }

            }

        }

    }

    class DownloadEvent implements ActionListener{

        @Override

        public void actionPerformed(ActionEvent e) {

            if (!textfield.getText().equals("На ноль делить нельзя")){

                if (e.getActionCommand().equals("s")){

                    save\_text = textfield.getText();

                }

                if (e.getActionCommand().equals("g")){

                    if (textfield.getText().equals("0")){

                    textfield.setText(save\_text);

                    }

                else{

                    textfield.setText(textfield.getText() + save\_text);

                }

                }

            }

        }

    }

}

**Результат:**

