ФГБОУ ВО

ННГАСУ

Кафедра информационных систем и технологий

Расчетно-графическая работа

Язык программирования Java

**Разработка графического интерфейса пользователя на Java**

Выполнила студентка 3 курса группы ИС-29

Хазова А.С. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Проверил старший преподаватель

Лахов А. Я. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Нижний Новгород

2023 год

**Содержание**

[Задания. 8 Вариант 3](#_Toc133924034)

[Задание 1. 3](#_Toc133924035)

[Задание 2. 3](#_Toc133924036)

[Проект графического интерфейса пользователя 4](#_Toc133924037)

[Диаграммы классов, разрабатываемого приложения 7](#_Toc133924038)

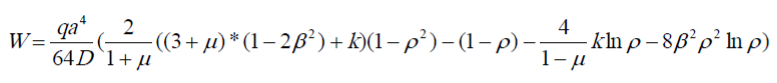
[Код на языке Java 8](#_Toc133924039)

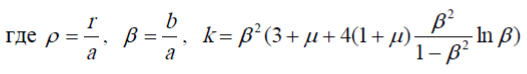
[Сеанс работы с приложением 25](#_Toc133924040)

# Задания. 8 Вариант

## Задание 1.

Составить программу для вычисления функции W в 12 равноотстоящих точках оси r (b < r < a) на языке Java.







## Задание 2.

Информация о среднесуточной температуре воздуха за месяц задана в виде массива t1, t2, t3, …, t30. Определить сколько дней температура была выше 10оC. Составить программу решения задачи.

# Проект графического интерфейса пользователя

Создан с помощью Microsoft Visio.

1. Главное окно приложения.

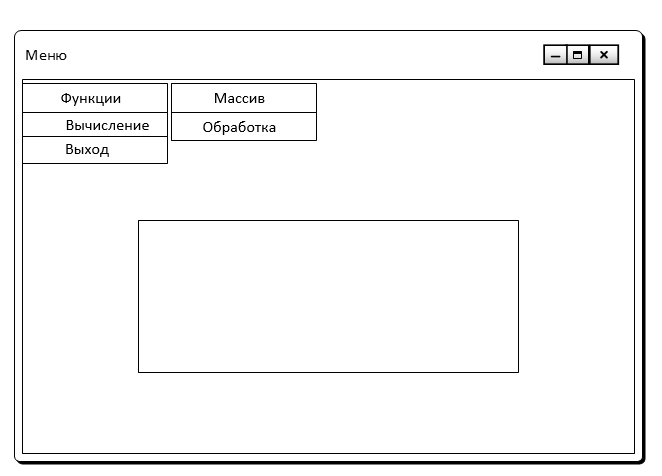


Рис. 1 – Проект главного окна разрабатываемого приложения.

1. Окно расчета функции.

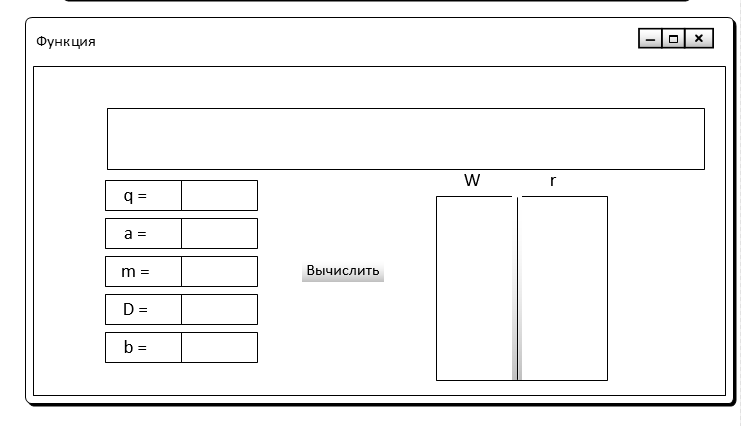


Рис.2 – Проект окна «Вычисление функции»

1. Окно работы с массивом.

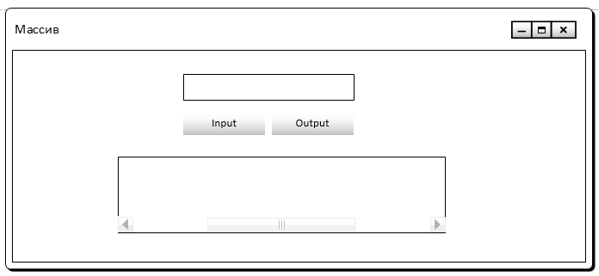


Рис.3 – Проект окна «Массив»

1. Окно ввода элемента массива.

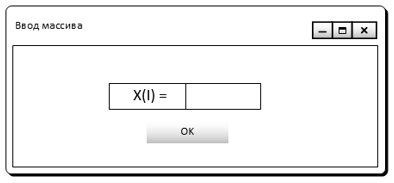


Рис.4 – Проект окна «Ввод массива»

1. Окно вывода результата.

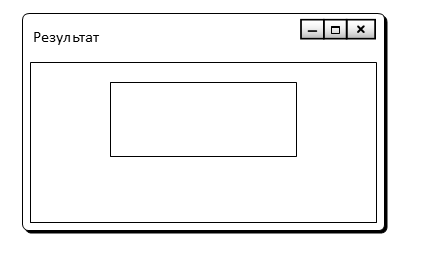


Рис.5 – Проект окна «Результат»

# Диаграммы классов, разрабатываемого приложения

Для реализации использовалась статическая структура UML.

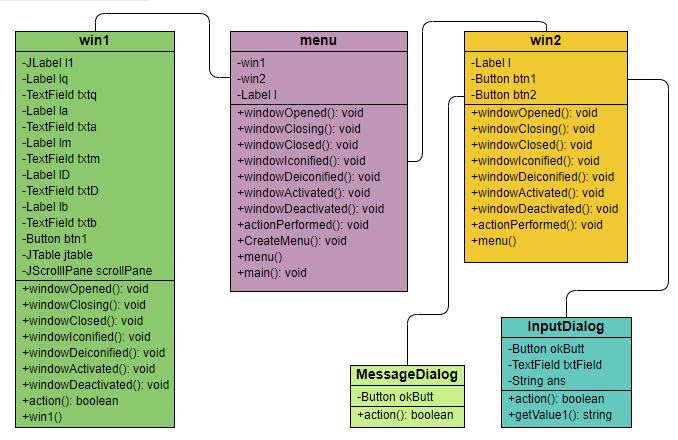


Рис.6 – Диаграмма классов разрабатываемого приложения

# Код на языке Java

# import java.awt.\*;

# import java.awt.event.ActionEvent;

# import java.awt.event.ActionListener;

# import java.awt.event.WindowEvent;

# import java.awt.event.WindowListener;

# import javax.swing.JLabel;

# import javax.swing.JTable;

# import javax.swing.JPanel;

# import javax.swing.JScrollPane;

# import java.lang.Math;

# import java.applet.\*;

# class MessageDialog extends Dialog {

# protected Button okButt;

# public MessageDialog(Frame parent, int k) {

# super(parent, true);

# setSize(230, 150);

# okButt = new Button("OK");

# JLabel messageLabel = new JLabel("Результат: ");

# Label res = new Label();

# if (k != 0) {

# messageLabel.setText("<html>Количество дней, когда <br>температура больше 10 градусов</html>");

# res.setText(" равно " + Integer.toString(k));

# } else {

# messageLabel.setText("Нет таких дней,");

# res.setText("когда температура больше 10");

# }

# setLayout(new FlowLayout());

# add(messageLabel);

# add(res);

# add(okButt);

# }

# public boolean action(Event evt, Object whichAction) {

# if (evt.target == okButt) {

# hide();

# }

# return true;

# }

# }

# class InputDialog extends Dialog {

# protected Button okButt;

# TextField txtField;

# String ans = "undef";

# public InputDialog(Frame parent, int i) {

# super(parent, true);

# setSize(200, 150);

# okButt = new Button("OK");

# Label messageLabel = new Label("Ввод X(" + i + ")");

# txtField = new TextField(18);

# setLayout(new FlowLayout());

# add(messageLabel);

# add(txtField);

# add(okButt);

# }

# public boolean action(Event evt, Object whichAction) {

# if (evt.target == okButt) {

# String sI = txtField.getText();

# ans = sI;

# hide();

# }

# return true;

# }

# public String getValue1() {

# return ans;

# }

# }

# class win1 extends Frame implements WindowListener {

# JLabel l1 = new JLabel(

# "<html>W = (q\*a<sup>4</sup>/64\*D)\*((2/1+m)\*((3+m)\*(1-2\*<sup>2</sup>)+k)\*(1-p<sup>2</sup>)-(1-p)-(4/1-m)\*k\*ln(p)-8\*B<sup>2</sup>\*p<sup>2</sup>\*ln(p)<br>p=r/a<br>B=b/a<br>k=B<sup>2</sup>\*(3+m+4\*(1+m)\*(B<sup>2</sup>/(1-B<sup>2</sup>)\*ln(B)</html>");

# Label lq = new Label("q =", Label.LEFT);

# TextField txtq = new TextField(11);

# Label la = new Label("a =", Label.LEFT);

# TextField txta = new TextField(11);

# Label lm = new Label("m =", Label.LEFT);

# TextField txtm = new TextField(11);

# Label lD = new Label("D =", Label.LEFT);

# TextField txtD = new TextField(11);

# Label lb = new Label("b =", Label.LEFT);

# TextField txtb = new TextField(11);

# Button btn1 = new Button("Вычислить");

# JTable jtable = new JTable();

# JScrollPane scrollPane = new JScrollPane();

# String[] columnNames = {

# "r",

# "W"

# };

# String[][] data = new String[12][4];

# public void windowOpened(WindowEvent arg0) {

# // TODO Auto-generated method stub

# }

# public void windowClosing(WindowEvent arg0) {

# String w = arg0.toString();

# l1.setText(w);

# dispose();

# }

# public void windowClosed(WindowEvent arg0) {

# // TODO Auto-generated method stub

# }

# public void windowIconified(WindowEvent arg0) {

# // TODO Auto-generated method stub

# }

# public void windowDeiconified(WindowEvent arg0) {

# // TODO Auto-generated method stub

# }

# public void windowActivated(WindowEvent arg0) {

# // TODO Auto-generated method stub

# }

# public void windowDeactivated(WindowEvent arg0) {

# // TODO Auto-generated method stub

# }

# public boolean action(Event evt, Object obj) {

# if (evt.target == btn1) {

# Double W, p, B, k, q, a, D, r, b, m;

# q = Double.valueOf(txtq.getText());

# a = Double.valueOf(txta.getText());

# m = Double.valueOf(txtm.getText());

# D = Double.valueOf(txtD.getText());

# b = Double.valueOf(txtb.getText());

# Double r1 = b + 0.0;

# r = (a - b) / 12;

# for (int i = 0; i <= 11; i++) {

# r1 += r;

# p = r1 / a;

# B = b / a;

# k = Math.pow(B, 2) \* (3 + m + 4 \* (1 + m) \* (Math.pow(B, 2) / (1 - Math.pow(B, 2))) \* Math.log(B));

# W = (q \* Math.pow(a, 4) / (64 \* D))

# \* ((2 / (1 + m)) \* ((3 + m) \* (1 - 2 \* Math.pow(B, 2)) + k) \* (1 - Math.pow(p, 2)) - (1 - p)

# - ((4 / (1 - m)) \* k \* Math.log(p))

# - (8 \* Math.pow(B, 2) \* Math.pow(p, 2) \* Math.log(p)));

# data[i] = new String[] { String.format("%.2f", r1),

# String.format("%.7f", W) };

# }

# JTable jtable = new JTable(data, columnNames);

# jtable.setPreferredScrollableViewportSize(new Dimension(200, 195));

# jtable.setFillsViewportHeight(true);

# scrollPane.setViewportView(jtable);

# this.setVisible(true);

# scrollPane.repaint();

# return true;

# } else

# return false;

# }

# public win1() {

# super("Вычисление функции");

# addWindowListener(this);

# setAlwaysOnTop(true);

# setSize(570, 350);

# setLayout(new FlowLayout(FlowLayout.CENTER));

# JPanel txtPanel = new JPanel();

# JPanel Panel1 = new JPanel();

# JPanel Panel2 = new JPanel();

# JPanel Panel3 = new JPanel();

# JPanel Panel4 = new JPanel();

# JPanel Panel5 = new JPanel();

# Panel1.add(lq);

# Panel1.add(txtq);

# Panel2.add(la);

# Panel2.add(txta);

# Panel3.add(lm);

# Panel3.add(txtm);

# Panel4.add(lD);

# Panel4.add(txtD);

# Panel5.add(lb);

# Panel5.add(txtb);

# add(l1);

# txtPanel.add(Panel1, BorderLayout.AFTER\_LAST\_LINE);

# txtPanel.add(Panel2, BorderLayout.AFTER\_LAST\_LINE);

# txtPanel.add(Panel3, BorderLayout.AFTER\_LAST\_LINE);

# txtPanel.add(Panel4, BorderLayout.AFTER\_LAST\_LINE);

# txtPanel.add(Panel5, BorderLayout.AFTER\_LAST\_LINE);

# txtPanel.setPreferredSize(new Dimension(200, 200));

# add(txtPanel);

# add(btn1);

# jtable.setPreferredScrollableViewportSize(new Dimension(200, 195));

# jtable.setFillsViewportHeight(true);

# scrollPane.setViewportView(jtable);

# this.setVisible(true);

# scrollPane.repaint();

# add(scrollPane);

# show();

# }

# }

# class win2 extends Frame implements WindowListener {

# Label l1 = new Label("Обработка массива из 30-ти элементов", Label.CENTER);

# TextArea txt = new TextArea(2, 35);

# Button btn1 = new Button("Input");

# Button btn2 = new Button("Output");

# int[] a = new int[30];

# public void windowOpened(WindowEvent arg0) {

# // TODO Auto-generated method stub

# }

# public void windowClosing(WindowEvent arg0) {

# String w = arg0.toString();

# l1.setText(w);

# dispose();

# }

# public void windowClosed(WindowEvent arg0) {

# // TODO Auto-generated method stub

# }

# public void windowIconified(WindowEvent arg0) {

# // TODO Auto-generated method stub

# }

# public void windowDeiconified(WindowEvent arg0) {

# // TODO Auto-generated method stub

# }

# public void windowActivated(WindowEvent arg0) {

# // TODO Auto-generated method stub

# }

# public void windowDeactivated(WindowEvent arg0) {

# // TODO Auto-generated method stub

# }

# public boolean action(Event evt, Object obj) {

# if (evt.target == btn1) {

# int i = 0;

# while (i < 30) {

# InputDialog ok = new InputDialog(this, i);

# ok.show();

# a[i] = Integer.parseInt(ok.getValue1());

# i++;

# }

# for (int j = 0; j < 30; j++) {

# txt.append(Integer.toString(a[j]) + " ");

# }

# return true;

# } else if (evt.target == btn2) {

# int k = 0;

# for (int i = 0; i < 30; i++) {

# if (a[i] > 10) {

# k += 1;

# }

# }

# Dialog ok = new MessageDialog(this, k);

# ok.show();

# return true;

# } else

# return false;

# }

# public win2() {

# super("Работа с массивом");

# addWindowListener(this);

# setAlwaysOnTop(true);

# setSize(300, 200);

# setLayout(new BorderLayout());

# add(l1);

# setLayout(new FlowLayout());

# btn1.setLocation(100, 100);

# btn1.setSize(50, 50);

# add(btn1);

# btn2.setLocation(100, 200);

# btn2.setSize(50, 50);

# add(btn2);

# add(txt);

# show();

# }

# }

# public class menu extends Frame implements WindowListener, ActionListener {

# Label l = new Label("Нажмите на меню", Label.CENTER);

# JLabel jlbl = new JLabel(

# "<html>Расчетно-графическая работа<br>Разработка графического интерфейса на языке программирования Java<br>Хазовой Александры Сергеевны<br>группы ИС-29 Вариант 8<br>2023 год<br>Задание 1.<br>Составить программу для вычисления функции W в 12 равноотстоящих<br> точках оси r (b&#60r&#60a).<br>Задание 2.<br>Информация о среднесуточной температуре воздуха за месяц задана<br>в виде массива t<sub>1</sub>, t<sub>2</sub>, ..., t<sub>29</sub>, t<sub>30</sub>.<br>Определить сколько дней температура была выше 10<sup>o</sup>C.</html>");

# public void windowOpened(WindowEvent arg0) {

# // TODO Auto-generated method stub

# }

# public void windowClosing(WindowEvent arg0) {

# dispose();

# System.exit(0);

# }

# public void windowClosed(WindowEvent arg0) {

# // TODO Auto-generated method stub

# }

# public void windowIconified(WindowEvent arg0) {

# // TODO Auto-generated method stub

# }

# public void windowDeiconified(WindowEvent arg0) {

# // TODO Auto-generated method stub

# }

# public void windowActivated(WindowEvent arg0) {

# // TODO Auto-generated method stub

# }

# public void windowDeactivated(WindowEvent arg0) {

# // TODO Auto-generated method stub

# }

# public void actionPerformed(ActionEvent arg0) {

# String cwd = arg0.getActionCommand();

# String str = "";

# if (cwd.equals("Вычисление")) {

# l.setText("Вычисление");

# new win1();

# show();

# } else if (cwd.equals("Обработка")) {

# l.setText("Обработка");

# new win2();

# show();

# } else if (cwd.equals("Выход")) {

# dispose();

# System.exit(0);

# }

# }

# void CreateMenu() {

# Menu m = new Menu("Функции");

# m.addActionListener(this);

# m.add(new MenuItem("Вычисление"));

# m.add(new MenuItem("Выход"));

# MenuBar mbar = new MenuBar();

# mbar.add(m);

# Menu m1 = new Menu("Массив");

# m1.addActionListener(this);

# m1.add(new MenuItem("Обработка"));

# mbar.add(m1);

# setMenuBar(mbar);

# }

# public menu() {

# super("РГР Java");

# CreateMenu();

# addWindowListener(this);

# setSize(500, 330);

# setLayout(new FlowLayout(FlowLayout.CENTER));

# add(l);

# add(jlbl);

# show();

# }

# public static void main(String[] args) {

# new menu(); }

# }

# Сеанс работы с приложением

1. Работа с главным окном приложения.

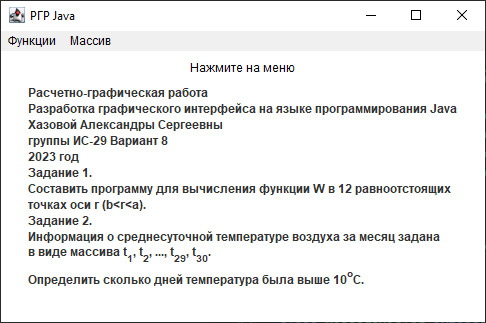
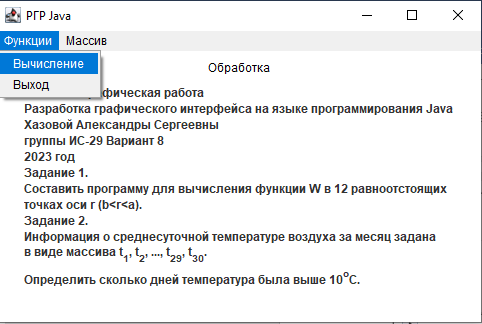


Рис.7 – Вид главного окна



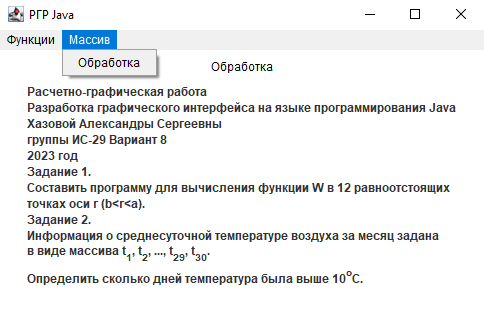


Рис.8-9 – Вид элементов меню

1. Работа с окном «Вычисление функции». Окно появляется при выборе элемента «Вычисление» в подменю «Функции» главного окна.

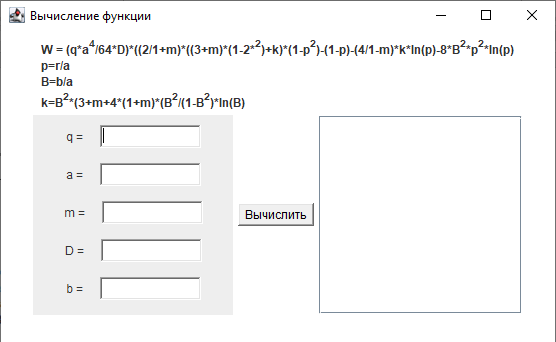


Рис.10 – Вид формы «Вычисление функции»

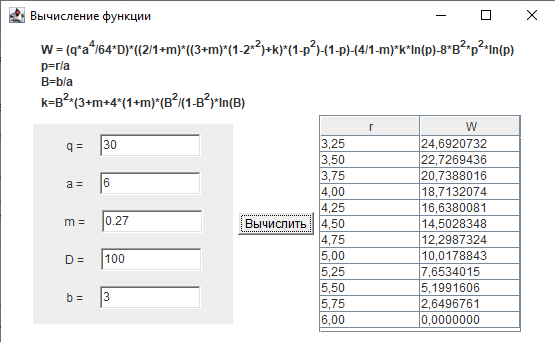


Рис.11 – Вид окна «Вычисление функции» с заполненными значениями переменных и выводом результатов вычислений в таблицу

1. Работа с окном «Работа с массивом». Окно вызывается при нажатии на элемент «Обработка» подменю «Массив» в главном окне.

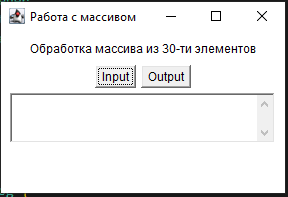


Рис.12 – Вид формы до ввода массива через кнопку Input

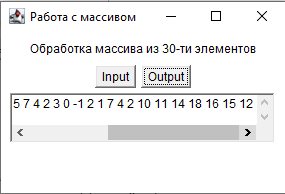
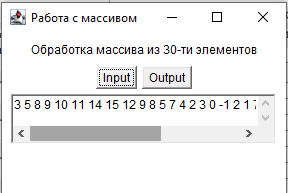


Рис.13-14 – Вид окна после ввода массива

1. Работа с формой, вызов которой осуществляется после нажатия на кнопку Input в окне «Работа с массивом».

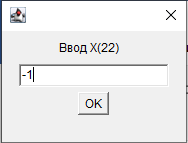
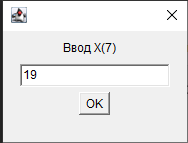
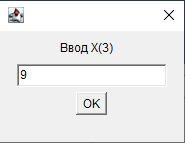


Рис.15-17 – Пример ввода массива через диалоговое окно

1. Работа с окном, вызов которого происходит при нажатии на кнопку Output в окне «Работа с массивом».

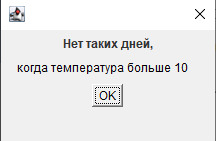
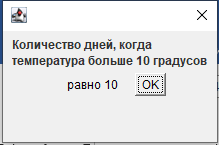


Рис. 18-19 – Варианты результата обработки массива и вывода их в отдельное окно приложения