

Содержание

[Введение 5](#_Toc136469387)

[1 Постановка задачи 6](#_Toc136469388)

[2 Выбор решения 7](#_Toc136469389)

[3 Описание программы 8](#_Toc136469390)

[3.1 Серверная часть 8](#_Toc136469391)

[3.2 Клиентская часть 8](#_Toc136469392)

[4 Описание способа организации пользовательского интерфейса 9](#_Toc136469393)

[5 Описание результатов работы программы 10](#_Toc136469394)

[5.1 Запуск программы 10](#_Toc136469395)

[5.2 Ввод значений и построение графика 11](#_Toc136469396)

[5.3 Сохранение и чтение графика функции 12](#_Toc136469397)

[Заключение 14](#_Toc136469398)

[Список используемой литературы 15](#_Toc136469399)

[Приложение А. Исходный код 16](#_Toc136469400)

[Приложение А.1 – Исходный код сервера 16](#_Toc136469401)

[Приложение А.1.1 – server.java 16](#_Toc136469402)

[Приложение А.1.2 – MathParser.java 17](#_Toc136469403)

[Приложение А.2 – Исходный код клиента 18](#_Toc136469404)

[Приложение А.2.1 – mainForm.java 18](#_Toc136469405)

[Приложение А.2.2 – GraphPanel.java 26](#_Toc136469406)

[Приложение B. UML-диаграммы приложения. 28](#_Toc136469407)

[Приложение B.1 – UML-диаграмма вариантов использования 28](#_Toc136469408)

[Приложение В.2 – UML-диаграмма классов клиента 29](#_Toc136469409)

[Приложение В.3 – UML-диаграмма классов сервера 30](#_Toc136469410)

[Приложение В.4 – UML-диаграмма деятельности 30](#_Toc136469411)

[Приложение В.5 – UML-диаграмма развёртывания 31](#_Toc136469412)

[Приложение В.6 – UML-диаграмма последовательности 31](#_Toc136469413)

# Введение

Java – широко используемый язык программирования для написания интернет-приложений. Язык Java широко использовался на протяжении более двух десятилетий. Миллионы приложений Java используются и сегодня. Java – это многоплатформенный, объектно-ориентированный и сетецентрический язык, который сам по себе может использоваться как платформа. Это быстрый, безопасный и надежный язык программирования для всего: от мобильных приложений и корпоративного ПО до приложений для работы с большими данными и серверных технологий.

Все языки программирования являются средством общения с машинами. Аппаратное обеспечение машины реагирует только на электронную связь. Языки программирования высокого уровня, такие как Java, играют роль моста между человеческим и аппаратным языком. Для использования Java разработчики должны понимать две вещи.

1. Язык Java и API — это внешний интерфейс между разработчиком и платформой Java.

2. Виртуальная машина Java — это внутренняя связь между платформой Java и базовым аппаратным оборудованием.

Представленная в данном курсовом проекте программа является простым клиент-серверным приложением, которое позволяет пользователю вводить математическую функцию и на её основе строить график.

# 1 Постановка задачи

Основной задачей данного курсового проекта является закрепление знаний и приобретение практических навыков разработки клиент-серверных приложений на языке Java. Для этого необходимо выполнить программную реализацию модульного приложения «Построение графика функции» на языке Java в среде разработки NetBeans.

Программа должна предоставлять возможность пользователю вписать математическую функцию и интервалы по осям X и Y. После того как пользователь впишет все необходимые значения и нажмёт на кнопку «Result» программа должна показать пользователю график той математической функции, которую вписал пользователь.

В проекте предусмотрен графический интерфейс и используются следующие технологии:

1. Java Collections Framework
2. Механизм обработки исключительных ситуаций
3. Java Stream API
4. Java Multithreading
5. Java Socket и Java Server Socket

# 2 Выбор решения

Для реализации проекта используются:

* Объектно-ориентированный язык Java;
* Библиотека Swing;
* Библиотека Exp4j
* Пакеты, содержащие файлы ресурсов, пользовательский интерфейс;
* Java Stream API;
* IDE NetBeans 16;

Основной задачей курсового проекта являлась реализация взаимодействия клиента и сервера. Для выполнения данной задачи использовались TCP сокеты.

Взаимодействие между клиентом и сервером должно происходить по принципу “Запрос-Ответ”. В обоих случаях, что со стороны сервера, что и со стороны клиента использовался TCP протокол, который позволяет точно передавать большие объёмы данных.

Интерфейс пользователя реализован с помощью библиотеки Java Swing и содержит в функционале:

* Поддержка ввода текста с клавиатуры
* Сохранение графика в файл
* Чтение графика из файла

# 3 Описание программы

Программа, состоит из клиентской и серверной части. Сначала запускается сервер, который входит в состояние ожидания запроса клиента.

После чего устройствах запускается клиентская часть, которая по нажатию определенных кнопок, посылает определённый запросы к серверу.

3.1 Серверная часть

Server.java – главный файл серверного процесса, котором находится точка входа в программу, функция main(). MathParser.java – файл, в котором происходит парсинг математических функций.

При получении запроса сервер создаёт нить для обработки введённых пользователем данных. Метод run(), находящийся в нити, в котором происходит создание объекта класса MathParser и вызов метода fillList(). Данный метод заполняет ArrayList координатами пикселей, которые нужны для построения графика функции. Далее происходит отправка этого ArrayList’а клиенту.

3.2 Клиентская часть

mainForm.java – главный файл пользовательского процесса, который реализует интерфейс и функции основного окна программы. GraphForm.java – файл, в котором происходит отрисовка графика функции.

После ввода всех необходимых значений и нажатия кнопки «Построить», происходит отправка всех необходимых данных на сервер. Сервер, обработав, данные возвращает клиенту ArrayList с координатами пикселей. Далее запускается нить, которая создаёт новый jFrame, добавляет в него объект класса GraphPanel, который наследуется от jPanel и на этой панели пользователь может увидеть график, введённой им, функции.

# 4 Описание способа организации пользовательского интерфейса

В качестве среды разработки была выбрана программа NetBeans. Программа предоставляет все средства, необходимые при разработке графического приложения.

Для реализации пользовательского интерфейса была использована библиотека Swing. Эта библиотека содержит более богатый и удобный набор элементов пользовательского интерфейса, обеспечивает одинаковое восприятие конечными пользователями приложений на разных платформах.

Основные кнопки для открытия файла, сохранения в файл, построения графика были использованы jButton.

Поля ввода функции и интервалов реализованы с помощью jTextField

# 5 Описание результатов работы программы

Среда разработки NetBeans предоставляет все средства, необходимые при разработке и отладке многомодульной программы.

Тестирование проводилось в рабочем порядке, в процессе разработки, после завершения написания программы. В ходе тестирования было выявлено и исправлено множество проблем, связанных с вводом данных, изменением дизайна клиентской части, алгоритмом программы.

Ниже продемонстрирован результат тестирования функционала программы.

* 1. Запуск программы

При запуске программы мы можем увидеть все элементы программы (рис. 1).

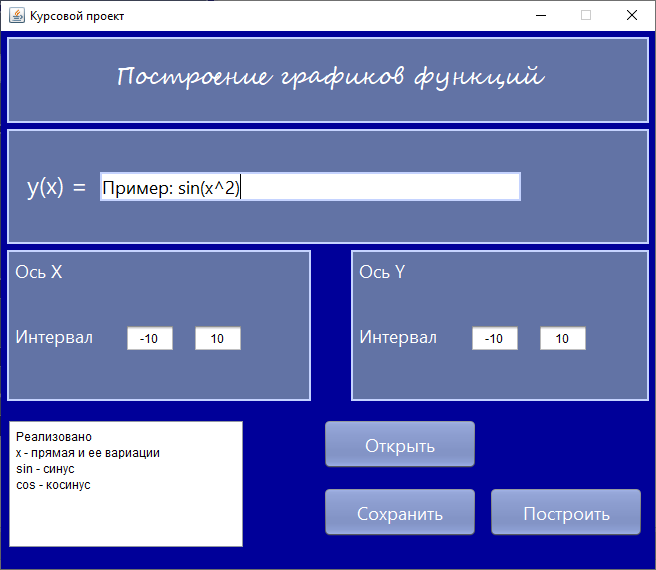


Рисунок – Запуск приложения

* 1. Ввод значений и построение графика

Для введения необходимых значений используются текстовые поля ввода данных.

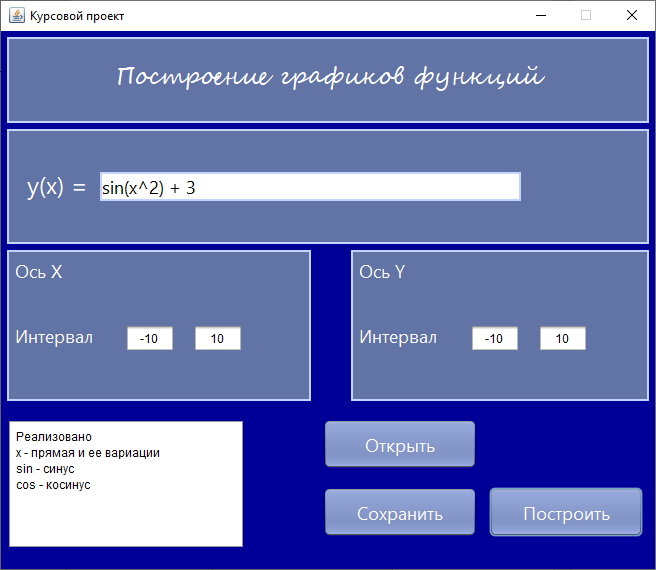


Рисунок – Ввод значений

При введении функции и интервалов по осям X и Y, и нажатии кнопки «Построить», откроется новое окно с графиком функции.

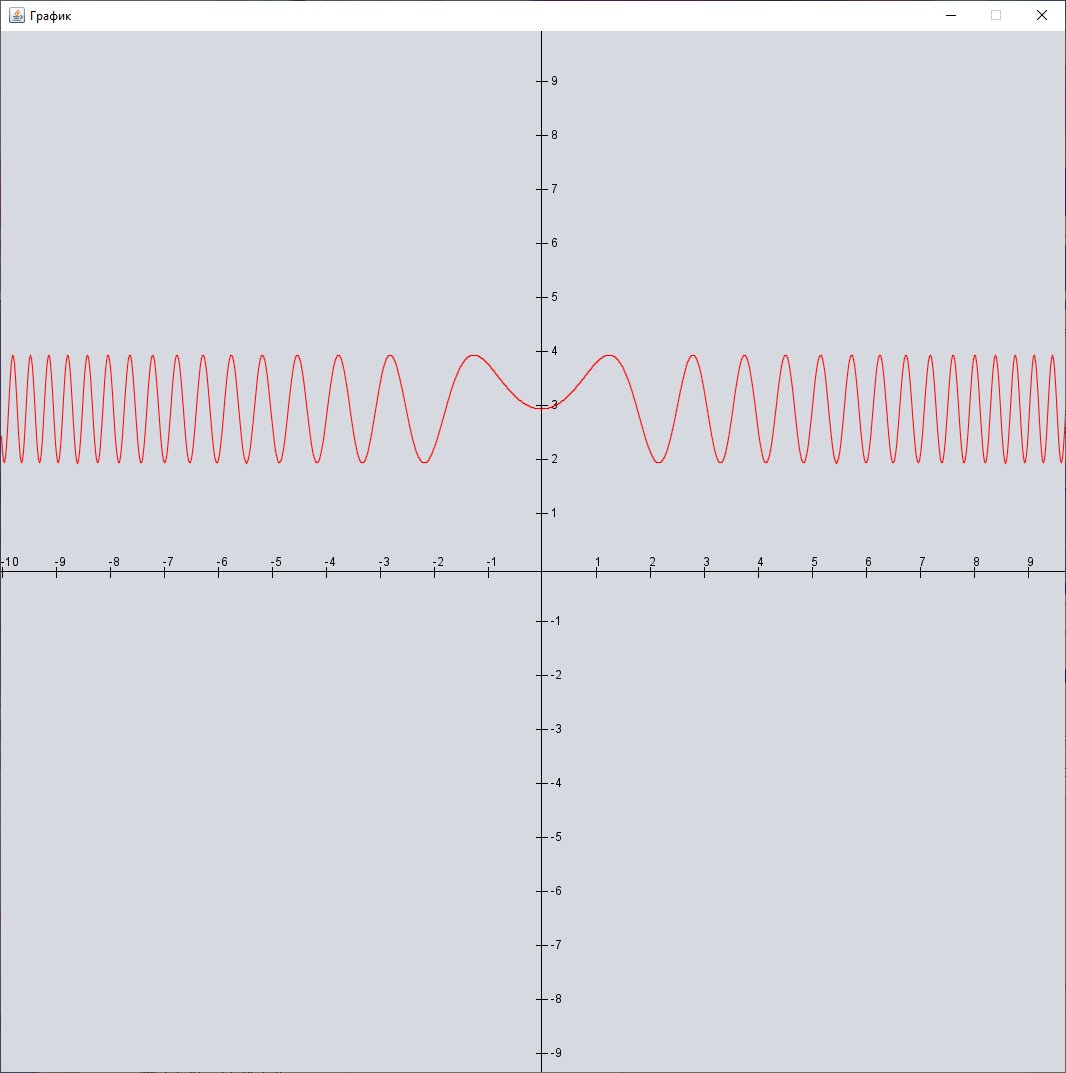


Рисунок – Построение графика функции

* 1. Сохранение и чтение графика функции

После построения графика пользователь может сохранить этот график в txt-файл, в котором храниться координаты пикселей графика, что позволяет пользователю, не обращаясь к серверу, отрисовать график функции.

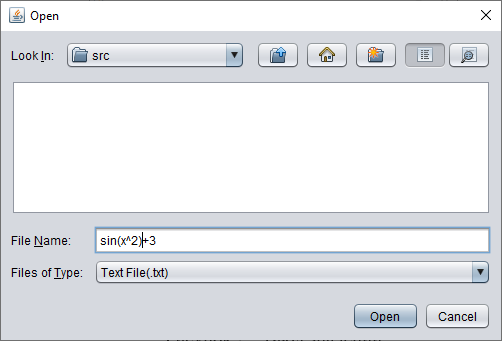


Рисунок – Сохранение графика

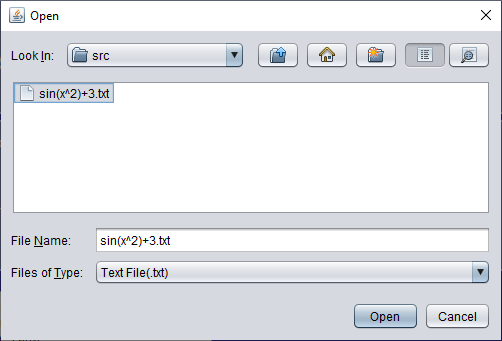


Рисунок – Загрузка графика

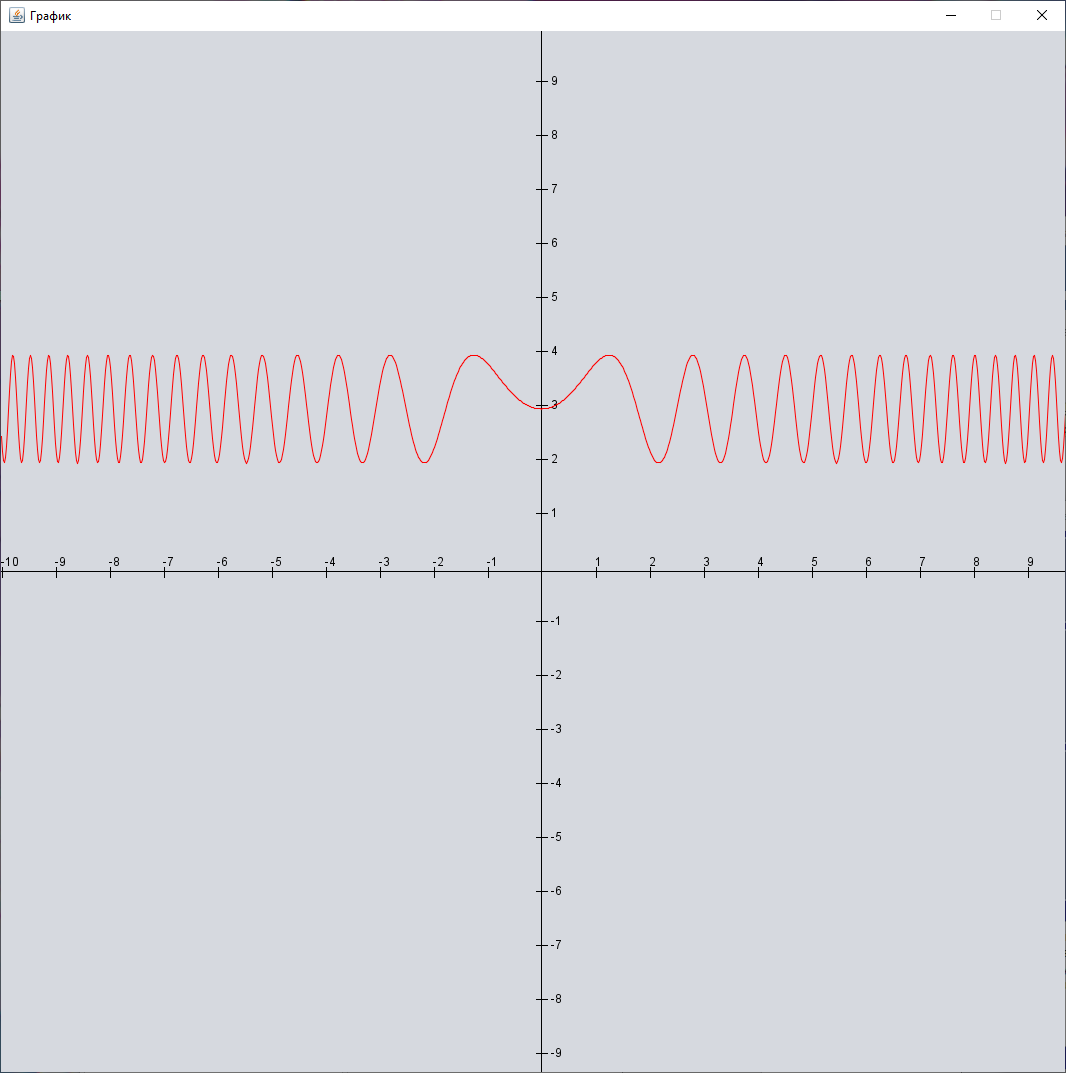


Рисунок – Результат работы чтения графика

# Заключение

В процессе выполнения данной курсовой работы я получил опыт разработки программ на высокоуровневом языке Java. Я овладел навыками создания клиент-серверных приложений с использованием сокетов и разработки графического интерфейса с помощью библиотеки Swing. Кроме того, я изучил возможности интегрированной среды разработки NetBeans.

В рамках этой работы я разработал программу, которая позволяет строить график математической функции, введённой пользователем.

# Список используемой литературы

1. Университет Информационных Технологий (ИНТУИТ), 2016.— 285с.
2. Портянкин, Java Swing: Эффектные пользовательские интерфейсы - Издание второе, 2011, 200 с.
3. Package javax.swing // docs.oracle.com : [сайт]. – 2023. – URL: https://docs.oracle.com/javase/8/docs/api/javax/swing/package-summary.html (дата обращения: 01.04.2023).
4. Шилдт, Г. Swing. Руководство для начинающих. – М.: Вильямс, 2007. – 704 с.: ил.
5. Берд, Барри Java для чайников / Барри Берд. - М.: Диалектика / Вильямс, 2013. - 521 c.
6. Дубаков А.А. Сетевое программирование: учебное пособие / А.А. Дубаков – СП: НИУ ИТМО, 2013. – 248 с.
7. Библиотека exp4j // www.objecthunter.net : [сайт]. – 2017. – URL:

https://www.objecthunter.net/exp4j/ (дата обращения 22.05.2023)

# Приложение А. Исходный код

Приложение А.1 – Исходный код сервера

Приложение А.1.1 – server.java

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.net.ServerSocket;

import java.net.Socket;

import java.util.ArrayList;

import java.util.logging.Level;

import java.util.logging.Logger;

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

/\*\*

\*

\* @author sdxop

\*/

public class Server {

static String Func;

static int WIDTH;

static int HEIGHT;

static int x;

static int y;

static ArrayList<Double> resList = new ArrayList();

public static class MyThread extends Thread {

ServerSocket server;

Socket socket;

MyThread(ServerSocket server, Socket socket) {

this.server = server;

this.socket = socket;

}

@Override

public void run() {

MathParser mp = new MathParser(Func, WIDTH, HEIGHT, x, y);

resList = mp.fillList();

ObjectOutputStream oos;

try {

oos = new ObjectOutputStream(socket.getOutputStream());

oos.writeObject(resList);

} catch (IOException ex) {

Logger.getLogger(Server.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

public static void main(String args[]) throws IOException, ClassNotFoundException {

ServerSocket server = new ServerSocket(6666);

Socket socket = server.accept();

while (true) {

String str = "?";

ObjectInputStream ois = new ObjectInputStream(socket.getInputStream());

str = (String) ois.readObject();

if (str != "?") {

//System.out.println(str);

String[] resArr = str.split(" ");

Func = resArr[0];

WIDTH = Integer.valueOf(resArr[1]);

HEIGHT = Integer.valueOf(resArr[2]);

x = Integer.valueOf(resArr[3]);

y = Integer.valueOf(resArr[4]);

MyThread myThread = new MyThread(server, socket);

myThread.start();

}

}

}

}

Приложение А.1.2 – MathParser.java

import java.util.ArrayList;

import net.objecthunter.exp4j.Expression;

import net.objecthunter.exp4j.ExpressionBuilder;

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

/\*\*

\*

\* @author sdxop

\*/

public class MathParser {

String Func;

int WIDTH;

int HEIGHT;

int x;

int y;

ArrayList<Double> resList = new ArrayList<>();

MathParser(String Func, int WIDTH, int HEIGHT, int x, int y) {

this.Func = Func;

this.WIDTH = WIDTH;

this.HEIGHT = HEIGHT;

this.x = x;

this.y = y;

}

public double execute(int nums, int x, int y) {

String result = Func;

result = result.replaceAll("\\s", "");

result = result.replaceAll("^(\\d+)([+-/])", "$1\*" + String.valueOf(y) + "$2");

result = result.replaceAll("(\\w|[)])([+-])(\\d+)(([+-])|$)", "$1\*" + String.valueOf(y) + "/s$2$3\*" + String.valueOf(y) + "$4");

result = result.replaceAll("(?![(])x\\^(\\d)(?![)])", "((x\*" + String.valueOf(y) + "/s)^($1))/" + String.valueOf(y) + "/(s)^($1-1)");

result = result.replaceAll("([+-/])(\\d+)([/])[A-Z]([/])([a-z])", "$1$2$3$5");

result = result.replaceAll("sin\\((\\w+)\\^(\\d+)\\)", "sin(($1/s)^$2)\*" + String.valueOf(y));

result = result.replaceAll("sin\\((\\w+)\\)", "sin($1/s)" + String.valueOf(y));

result = result.replaceAll("cos\\((\\w+)\\^(\\d+)\\)", "cos(($1/s)^$2)\*" + String.valueOf(y));

result = result.replaceAll("cos\\((\\w+)\\)", "cos($1/s)\*" + String.valueOf(y));

result = result.replaceAll("tg\\((\\w+)\\^(\\d+)\\)", "tan(($1/s)^$2)\*" + String.valueOf(y));

result = result.replaceAll("tg\\((\\w+)\\)", "((sin($1/s))/(cos($1/s)))\*" + String.valueOf(y));

Expression e = new ExpressionBuilder(result)

.variables("x", "s")

.build()

.setVariable("x", nums)

.setVariable("s", x);

double resultT = e.evaluate();

return resultT;

}

public ArrayList fillList() {

for (int i = -WIDTH / 2; i < WIDTH / 2; i++) {

resList.add(this.execute(i, x, y));

}

return resList;

}

}

Приложение А.2 – Исходный код клиента

Приложение А.2.1 – mainForm.java

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.net.Socket;

import java.util.ArrayList;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.\*;

import javax.swing.filechooser.FileNameExtensionFilter;

/\*\*

\*

\* @author sdxop

\*/

class FileTools {

String FileName;

public String Open() {

JFileChooser fileChooser = new JFileChooser();

fileChooser.setCurrentDirectory(new File("./src"));

FileNameExtensionFilter txt = new FileNameExtensionFilter("Text File(.txt)", "txt");

FileNameExtensionFilter bin = new FileNameExtensionFilter("Bin File(.bin)", "bin");

fileChooser.addChoosableFileFilter(txt);

fileChooser.setFileFilter(txt);

int response = fileChooser.showDialog(null, null);

if (response == JFileChooser.APPROVE\_OPTION) {

FileName = fileChooser.getSelectedFile().getAbsolutePath();

}

return FileName;

}

}

class Runner implements Runnable {

int X\_MIN, X\_MAX, Y\_MIN, Y\_MAX;

ArrayList<Double> resList = new ArrayList<>();

Runner(ArrayList resList, int xmin, int xmax, int ymin, int ymax){

this.resList = resList;

this.X\_MAX = xmax;

this.X\_MIN = xmin;

this.Y\_MAX = ymax;

this.Y\_MIN = ymin;

}

@Override

public void run() {

JFrame scene = new JFrame("График");

scene.add(new GraphForm(resList, X\_MIN, X\_MAX, Y\_MIN, Y\_MAX));

scene.setSize(1080, 1080);

scene.setResizable(false);

scene.setVisible(true);

}

}

public class mainForm extends javax.swing.JFrame {

Socket socket;

ArrayList<Double> resList = new ArrayList<>();

/\*\*

\* Creates new form mainForm

\*/

public mainForm() throws IOException {

socket = new Socket("localhost", 6666);

initComponents();

}

public class foo extends javax.swing.JFrame{

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel5 = new javax.swing.JPanel();

jPanel2 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jPanel4 = new javax.swing.JPanel();

Formula = new javax.swing.JTextField();

jLabel2 = new javax.swing.JLabel();

jPanel1 = new javax.swing.JPanel();

LabelOsX = new javax.swing.JLabel();

minX = new javax.swing.JTextField();

maxX = new javax.swing.JTextField();

LabelIntervalX = new javax.swing.JLabel();

jPanel3 = new javax.swing.JPanel();

LabelOsY = new javax.swing.JLabel();

minY = new javax.swing.JTextField();

maxY = new javax.swing.JTextField();

LabelIntervalY = new javax.swing.JLabel();

jScrollPane1 = new javax.swing.JScrollPane();

jTextArea1 = new javax.swing.JTextArea();

FillGraph = new javax.swing.JButton();

SaveFile = new javax.swing.JButton();

OpenFile = new javax.swing.JButton();

LabelOpenFile = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Курсовой проект");

setBackground(new java.awt.Color(0, 0, 135));

setLocation(new java.awt.Point(1000, 500));

setName("mainForm"); // NOI18N

setResizable(false);

jPanel5.setBackground(new java.awt.Color(0, 0, 153));

jPanel5.setNextFocusableComponent(Formula);

jPanel2.setBackground(new java.awt.Color(98, 115, 165));

jPanel2.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(197, 211, 255), 2));

jLabel1.setFont(new java.awt.Font("Segoe Script", 0, 24)); // NOI18N

jLabel1.setForeground(new java.awt.Color(255, 255, 255));

jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel1.setText("Построение графиков функций");

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(18, 18, 18)

.addComponent(jLabel1)

.addContainerGap(25, Short.MAX\_VALUE))

);

jPanel4.setBackground(new java.awt.Color(98, 115, 165));

jPanel4.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(197, 211, 255), 2));

Formula.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N

Formula.setText("Пример: sin(x^2)");

Formula.setToolTipText("");

Formula.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(197, 211, 255), 2));

Formula.setVerifyInputWhenFocusTarget(false);

jLabel2.setFont(new java.awt.Font("Segoe UI", 0, 24)); // NOI18N

jLabel2.setForeground(new java.awt.Color(255, 255, 255));

jLabel2.setText("y(x) = ");

javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);

jPanel4.setLayout(jPanel4Layout);

jPanel4Layout.setHorizontalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel4Layout.createSequentialGroup()

.addGap(18, 18, 18)

.addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(Formula, javax.swing.GroupLayout.PREFERRED\_SIZE, 421, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

jPanel4Layout.setVerticalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel4Layout.createSequentialGroup()

.addGap(37, 37, 37)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(Formula, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel2))

.addContainerGap(41, Short.MAX\_VALUE))

);

jPanel1.setBackground(new java.awt.Color(98, 115, 165));

jPanel1.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(197, 211, 255), 2));

jPanel1.setPreferredSize(new java.awt.Dimension(298, 148));

LabelOsX.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N

LabelOsX.setForeground(new java.awt.Color(255, 255, 255));

LabelOsX.setText("Ось X ");

minX.setHorizontalAlignment(javax.swing.JTextField.CENTER);

minX.setText("-10");

maxX.setHorizontalAlignment(javax.swing.JTextField.CENTER);

maxX.setText("10");

LabelIntervalX.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N

LabelIntervalX.setForeground(new java.awt.Color(255, 255, 255));

LabelIntervalX.setText("Интервал");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(LabelOsX)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(LabelIntervalX)

.addGap(32, 32, 32)

.addComponent(minX, javax.swing.GroupLayout.PREFERRED\_SIZE, 50, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(maxX, javax.swing.GroupLayout.PREFERRED\_SIZE, 50, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(66, Short.MAX\_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addComponent(LabelOsX)

.addGap(40, 40, 40)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(LabelIntervalX)

.addComponent(minX, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(maxX, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

jPanel3.setBackground(new java.awt.Color(98, 115, 165));

jPanel3.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(197, 211, 255), 2));

LabelOsY.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N

LabelOsY.setForeground(new java.awt.Color(255, 255, 255));

LabelOsY.setText("Ось Y ");

minY.setHorizontalAlignment(javax.swing.JTextField.CENTER);

minY.setText("-10");

maxY.setHorizontalAlignment(javax.swing.JTextField.CENTER);

maxY.setText("10");

LabelIntervalY.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N

LabelIntervalY.setForeground(new java.awt.Color(255, 255, 255));

LabelIntervalY.setText("Интервал");

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);

jPanel3.setLayout(jPanel3Layout);

jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel3Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(LabelOsY)

.addGroup(jPanel3Layout.createSequentialGroup()

.addComponent(LabelIntervalY)

.addGap(33, 33, 33)

.addComponent(minY, javax.swing.GroupLayout.PREFERRED\_SIZE, 50, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(maxY, javax.swing.GroupLayout.PREFERRED\_SIZE, 50, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(59, Short.MAX\_VALUE))

);

jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel3Layout.createSequentialGroup()

.addContainerGap()

.addComponent(LabelOsY)

.addGap(40, 40, 40)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(LabelIntervalY)

.addComponent(minY, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(maxY, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(47, Short.MAX\_VALUE))

);

jTextArea1.setEditable(false);

jTextArea1.setColumns(20);

jTextArea1.setRows(7);

jTextArea1.setText("Реализовано\nх - прямая и ее вариации\nsin - синус\ncos - косинус\n");

jScrollPane1.setViewportView(jTextArea1);

FillGraph.setBackground(new java.awt.Color(98, 115, 165));

FillGraph.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N

FillGraph.setForeground(new java.awt.Color(255, 255, 255));

FillGraph.setText("Построить");

FillGraph.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

FillGraphActionPerformed(evt);

}

});

SaveFile.setBackground(new java.awt.Color(98, 115, 165));

SaveFile.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N

SaveFile.setForeground(new java.awt.Color(255, 255, 255));

SaveFile.setText("Сохранить");

SaveFile.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

SaveFileActionPerformed(evt);

}

});

OpenFile.setBackground(new java.awt.Color(98, 115, 165));

OpenFile.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N

OpenFile.setForeground(new java.awt.Color(255, 255, 255));

OpenFile.setText("Открыть");

OpenFile.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

OpenFileActionPerformed(evt);

}

});

LabelOpenFile.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N

LabelOpenFile.setForeground(new java.awt.Color(255, 255, 255));

LabelOpenFile.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

javax.swing.GroupLayout jPanel5Layout = new javax.swing.GroupLayout(jPanel5);

jPanel5.setLayout(jPanel5Layout);

jPanel5Layout.setHorizontalGroup(

jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel5Layout.createSequentialGroup()

.addGap(0, 6, Short.MAX\_VALUE)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jPanel4, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(jPanel5Layout.createSequentialGroup()

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel5Layout.createSequentialGroup()

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 304, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED))

.addGroup(jPanel5Layout.createSequentialGroup()

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)))

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel5Layout.createSequentialGroup()

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(OpenFile, javax.swing.GroupLayout.PREFERRED\_SIZE, 154, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(SaveFile, javax.swing.GroupLayout.PREFERRED\_SIZE, 154, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(FillGraph, javax.swing.GroupLayout.DEFAULT\_SIZE, 154, Short.MAX\_VALUE)

.addComponent(LabelOpenFile, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(6, 6, 6))

.addComponent(jPanel3, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))))

.addContainerGap())

);

jPanel5Layout.setVerticalGroup(

jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel5Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(18, 18, 18)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, jPanel5Layout.createSequentialGroup()

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(OpenFile, javax.swing.GroupLayout.DEFAULT\_SIZE, 50, Short.MAX\_VALUE)

.addComponent(LabelOpenFile, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(18, 18, 18)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(SaveFile, javax.swing.GroupLayout.PREFERRED\_SIZE, 50, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(FillGraph, javax.swing.GroupLayout.PREFERRED\_SIZE, 50, javax.swing.GroupLayout.PREFERRED\_SIZE))))

.addGap(0, 20, Short.MAX\_VALUE))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(jPanel5, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel5, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

);

pack();

}// </editor-fold>

private void FillGraphActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

if(!LabelOpenFile.getText().isEmpty()){

LabelOpenFile.setText("");

Thread myThread = new Thread(new Runner(resList,Integer.parseInt(minX.getText()),Integer.parseInt(maxX.getText()),Integer.parseInt(minY.getText()),Integer.parseInt(maxY.getText())));

myThread.start();

}

else if (!Formula.getText().isEmpty()) {

String forml = Formula.getText().replaceAll("\\s", "");

String str = forml + " 1080 1080 " + String.valueOf(1080 / (Integer.valueOf(maxX.getText()) - Integer.valueOf(minX.getText()))) + " " + String.valueOf(1080 / (Integer.valueOf(maxY.getText()) - Integer.valueOf(minY.getText())));

try {

ObjectOutputStream oos = new ObjectOutputStream(socket.getOutputStream());

oos.writeObject(str);

} catch (IOException ex) {

Logger.getLogger(mainForm.class.getName()).log(Level.SEVERE, null, ex);

}

try {

ObjectInputStream ois = new ObjectInputStream(socket.getInputStream());

resList = (ArrayList<Double>) ois.readObject();

} catch (IOException ex) {

Logger.getLogger(mainForm.class.getName()).log(Level.SEVERE, null, ex);

} catch (ClassNotFoundException ex) {

Logger.getLogger(mainForm.class.getName()).log(Level.SEVERE, null, ex);

}

Thread myThread = new Thread(new Runner(resList,Integer.parseInt(minX.getText()),Integer.parseInt(maxX.getText()),Integer.parseInt(minY.getText()),Integer.parseInt(maxY.getText())));

myThread.start();

}

}

private void SaveFileActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

try {

if (!resList.isEmpty()) {

FileTools ft = new FileTools();

String fileName = ft.Open() + ".txt";

FileOutputStream fos = new FileOutputStream(fileName);

ObjectOutputStream oos = new ObjectOutputStream(fos);

oos.writeObject(resList);

}

} catch (FileNotFoundException ex) {

Logger.getLogger(mainForm.class.getName()).log(Level.SEVERE, null, ex);

} catch (IOException ex) {

Logger.getLogger(mainForm.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void OpenFileActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

try {

FileTools ft = new FileTools();

String fileName = ft.Open();

if (fileName != null) {

FileInputStream fis = new FileInputStream(fileName);

ObjectInputStream ois = new ObjectInputStream(fis);

resList = (ArrayList) ois.readObject();

if (!resList.isEmpty()) {

LabelOpenFile.setText("Файл Открыт");

}

}

} catch (FileNotFoundException ex) {

Logger.getLogger(mainForm.class.getName()).log(Level.SEVERE, null, ex);

} catch (IOException ex) {

Logger.getLogger(mainForm.class.getName()).log(Level.SEVERE, null, ex);

} catch (ClassNotFoundException ex) {

Logger.getLogger(mainForm.class.getName()).log(Level.SEVERE, null, ex);

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(mainForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(mainForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(mainForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(mainForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

@Override

public void run() {

try {

new mainForm().setVisible(true);

} catch (IOException ex) {

Logger.getLogger(mainForm.class.getName()).log(Level.SEVERE, null, ex);

}

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton FillGraph;

private javax.swing.JTextField Formula;

private javax.swing.JLabel LabelIntervalX;

private javax.swing.JLabel LabelIntervalY;

private javax.swing.JLabel LabelOpenFile;

private javax.swing.JLabel LabelOsX;

private javax.swing.JLabel LabelOsY;

private javax.swing.JButton OpenFile;

private javax.swing.JButton SaveFile;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JPanel jPanel3;

private javax.swing.JPanel jPanel4;

private javax.swing.JPanel jPanel5;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTextArea jTextArea1;

private javax.swing.JTextField maxX;

private javax.swing.JTextField maxY;

private javax.swing.JTextField minX;

private javax.swing.JTextField minY;

// End of variables declaration

}

Приложение А.2.2 – GraphPanel.java

import java.awt.\*;

import java.awt.geom.Line2D;

import java.util.ArrayList;

import javax.swing.\*;

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

/\*\*

\*

\* @author sdxop

\*/

class GraphForm extends JPanel{

int HEIGHT, WIDTH;

int X\_MIN, X\_MAX, Y\_MIN, Y\_MAX;

String FORML;

ArrayList<Double> resList = new ArrayList<>();

GraphForm(ArrayList resList, int x\_min, int x\_max, int y\_min, int y\_max){

this.HEIGHT = 1080;

this.WIDTH = 1080;

this.resList = resList;

this.X\_MIN = x\_min;

this.X\_MAX = x\_max;

this.Y\_MIN = y\_min;

this.Y\_MAX = y\_max;

}

@Override

protected void paintComponent(Graphics g){

super.paintComponent(g);

Graphics2D G2D = (Graphics2D)g;

G2D.setRenderingHint(RenderingHints.KEY\_ANTIALIASING, RenderingHints.VALUE\_ANTIALIAS\_ON);

G2D.draw(new Line2D.Double(0, HEIGHT / 2, WIDTH, HEIGHT / 2));

G2D.draw(new Line2D.Double(WIDTH / 2, 0, WIDTH / 2, HEIGHT));

int step = 0;

//int s = WIDTH / (X\_MAX - X\_MIN);

for (int i = X\_MIN; i <= X\_MAX; i++) {

if (i != 0) {

G2D.drawString(String.valueOf(i), step, HEIGHT / 2 - 5);

G2D.drawString("|", step, HEIGHT / 2 + 5);

}

step += WIDTH / (X\_MAX - X\_MIN);

}

step = 0;

for (int i = Y\_MIN; i <= Y\_MAX; i++) {

if (i != 0) {

G2D.drawString(String.valueOf(-i), WIDTH / 2 + 10, step);

G2D.drawString("—", WIDTH / 2 - 5, step);

}

step += HEIGHT / (Y\_MAX - Y\_MIN);

}

G2D.setPaint(Color.RED);

for (int i = 0; i < resList.size() - 1; i++) {

G2D.draw(new Line2D.Double(i, -resList.get(i) + HEIGHT / 2, i + 1, -resList.get(i + 1) + HEIGHT / 2));

}

}

@FunctionalInterface

public interface Operation {

double execute(int num);

}

}

# Приложение B. UML-диаграммы приложения.

Приложение B.1 – UML-диаграмма вариантов использования

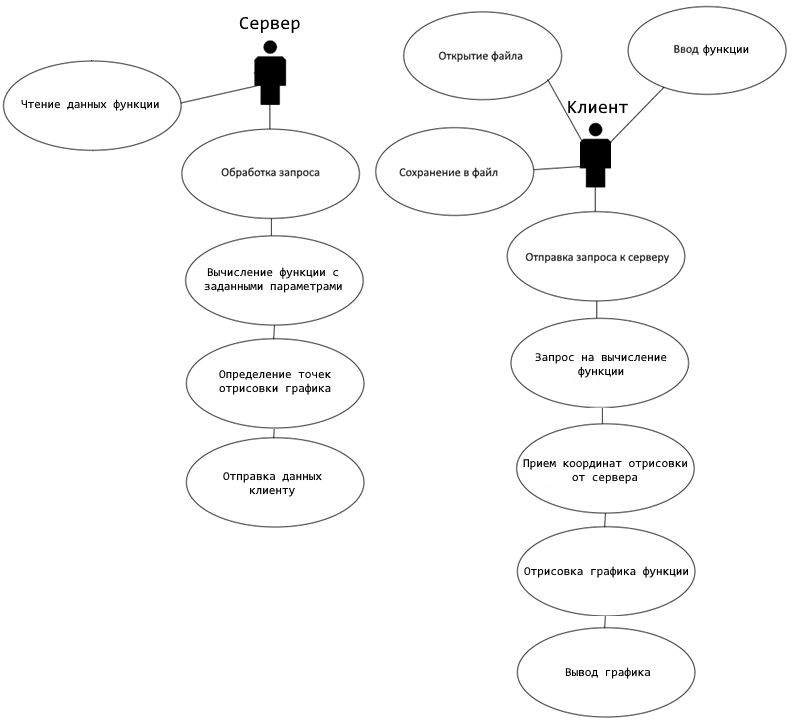


Рисунок – UML-диаграмма вариантов использования

Приложение В.2 – UML-диаграмма классов клиента

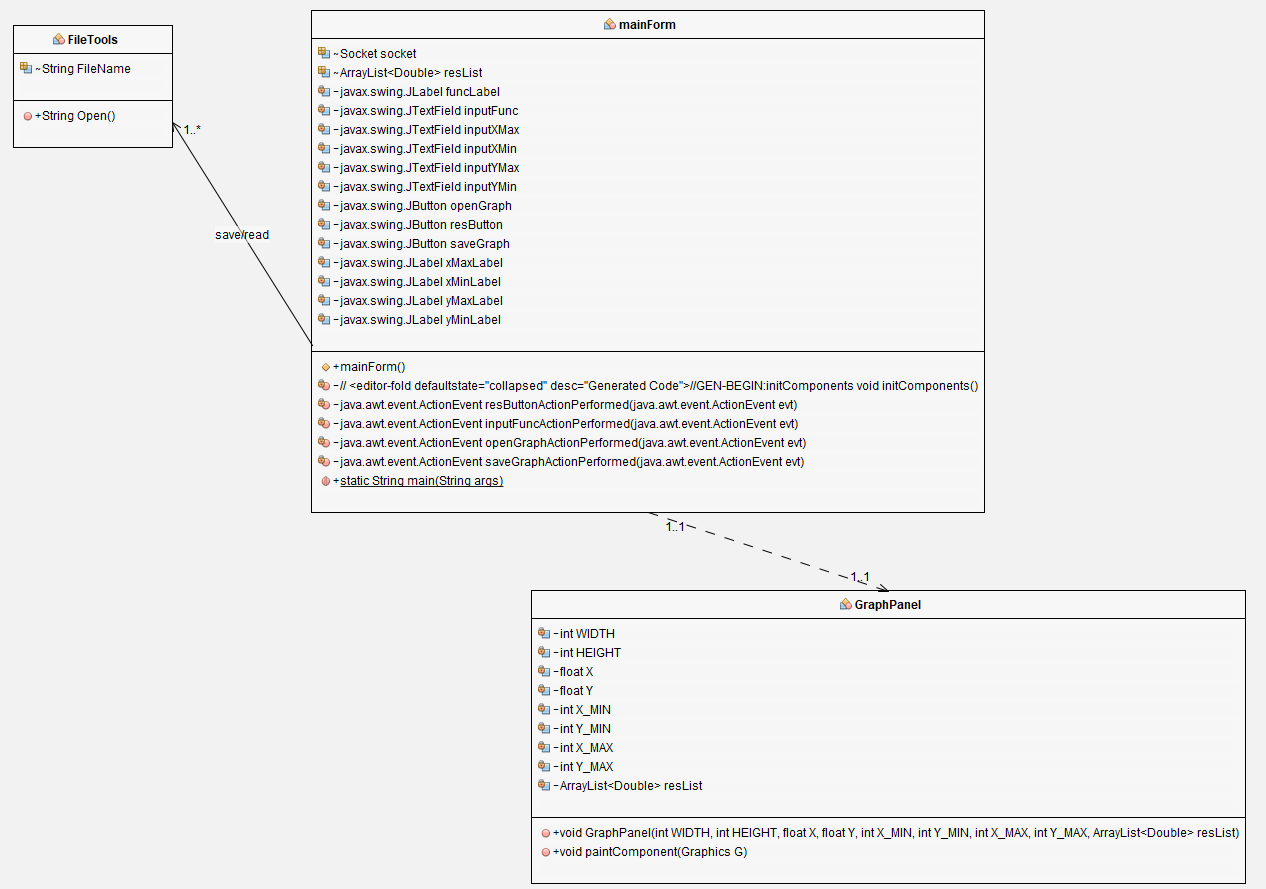


Рисунок – UML-диаграмма классов клиента

Приложение В.3 – UML-диаграмма классов сервера

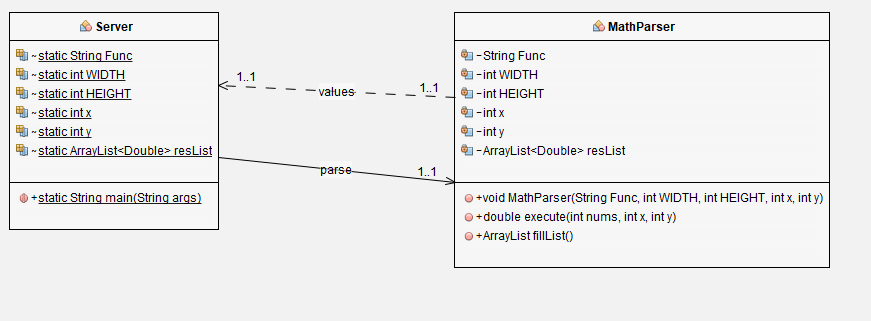


Рисунок – UML-диаграмма классов сервера

Приложение В.4 – UML-диаграмма деятельности

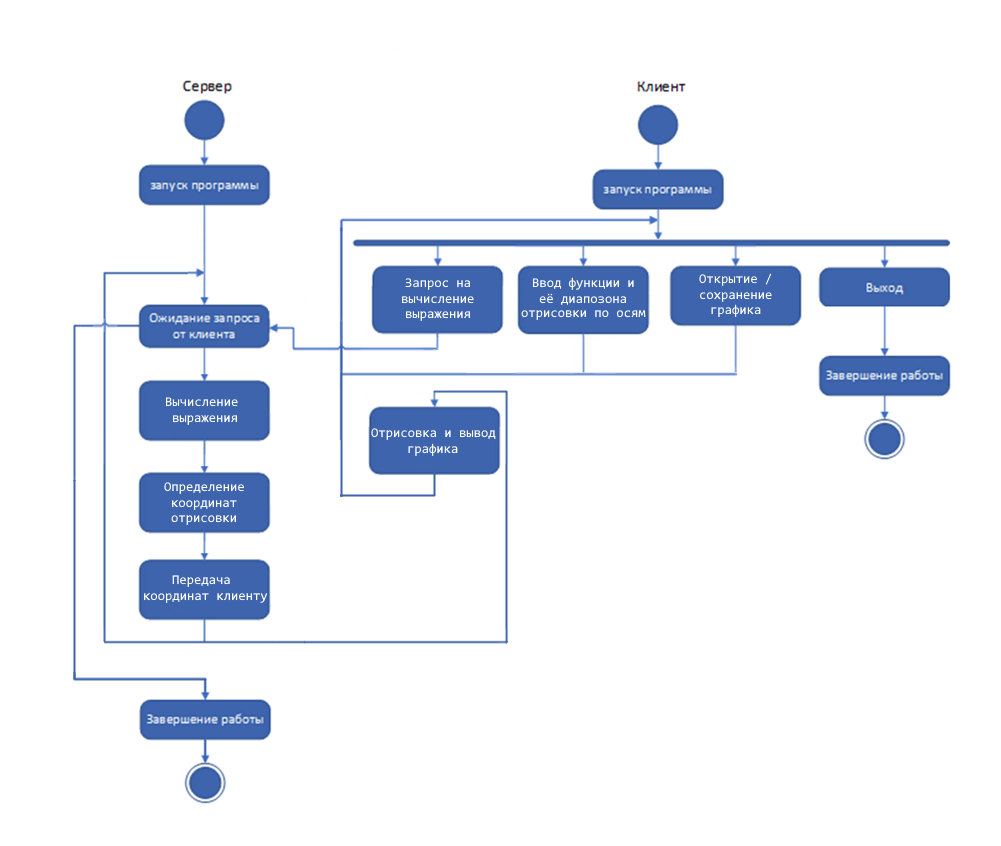


Рисунок – UML-диаграмма деятельности

Приложение В.5 – UML-диаграмма развёртывания

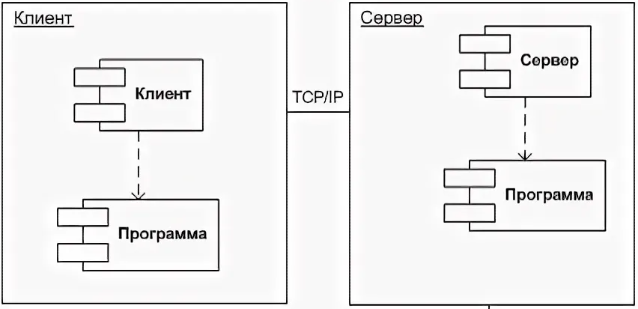


Рисунок – UML-диаграмма развёртывания

Приложение В.6 – UML-диаграмма последовательности

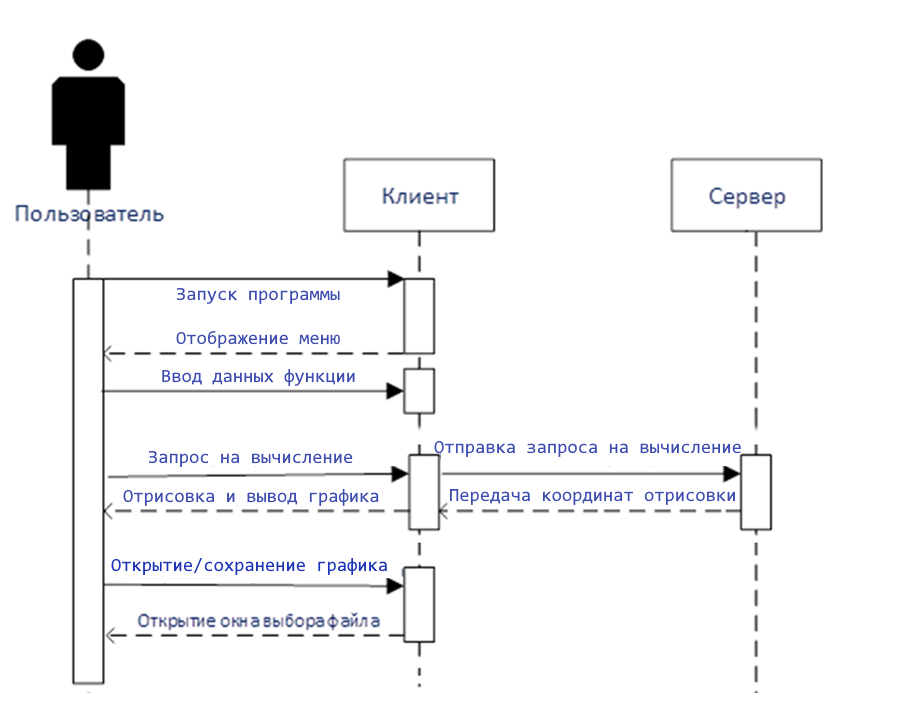


Рисунок – UML-диаграмма последовательности