Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**Пояснительная записка**

к курсовой работе

по курсу «Программирование на языке Java»

на тему «Разработка многомодульного приложения на языке Java»

Выполнил:

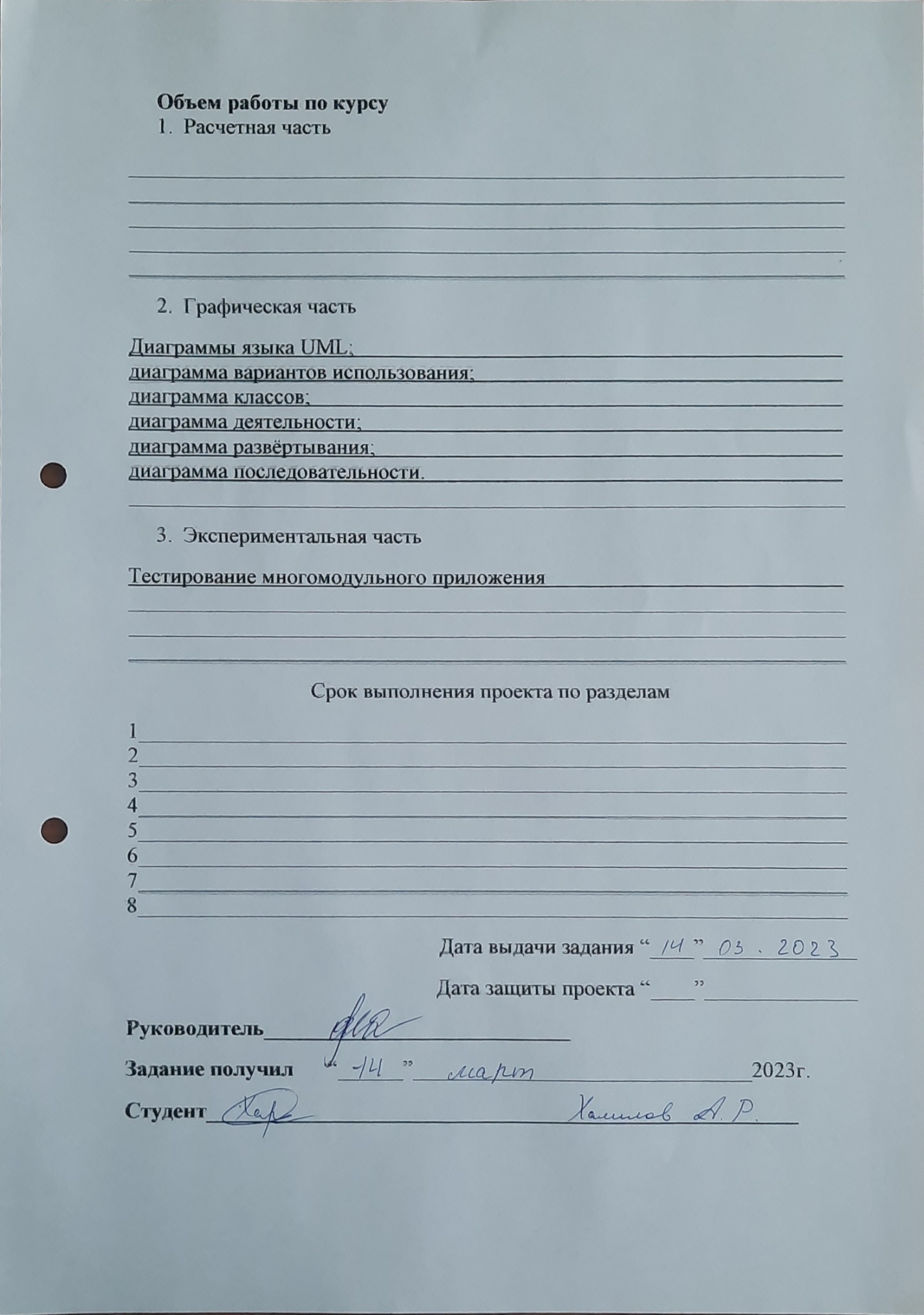
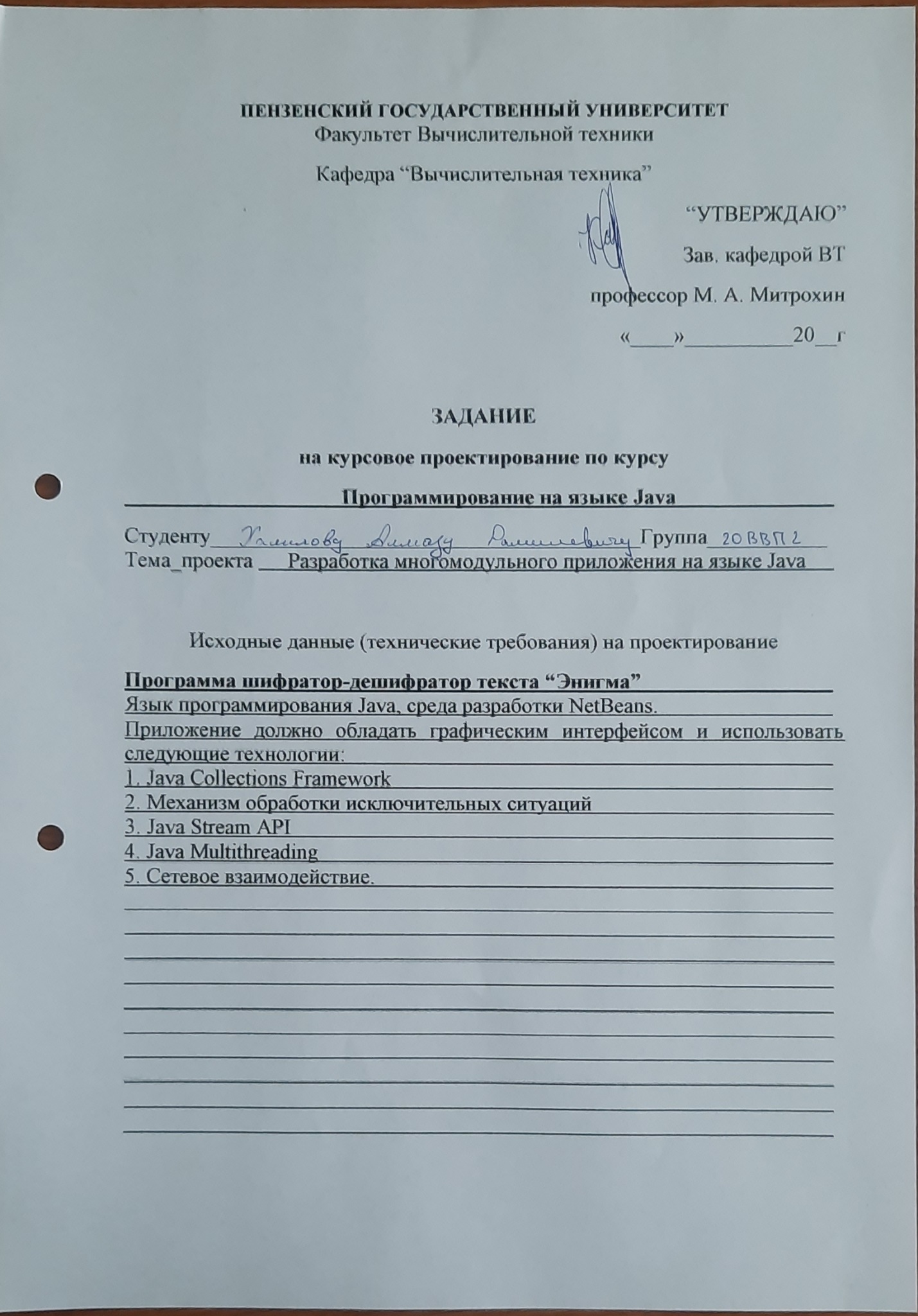
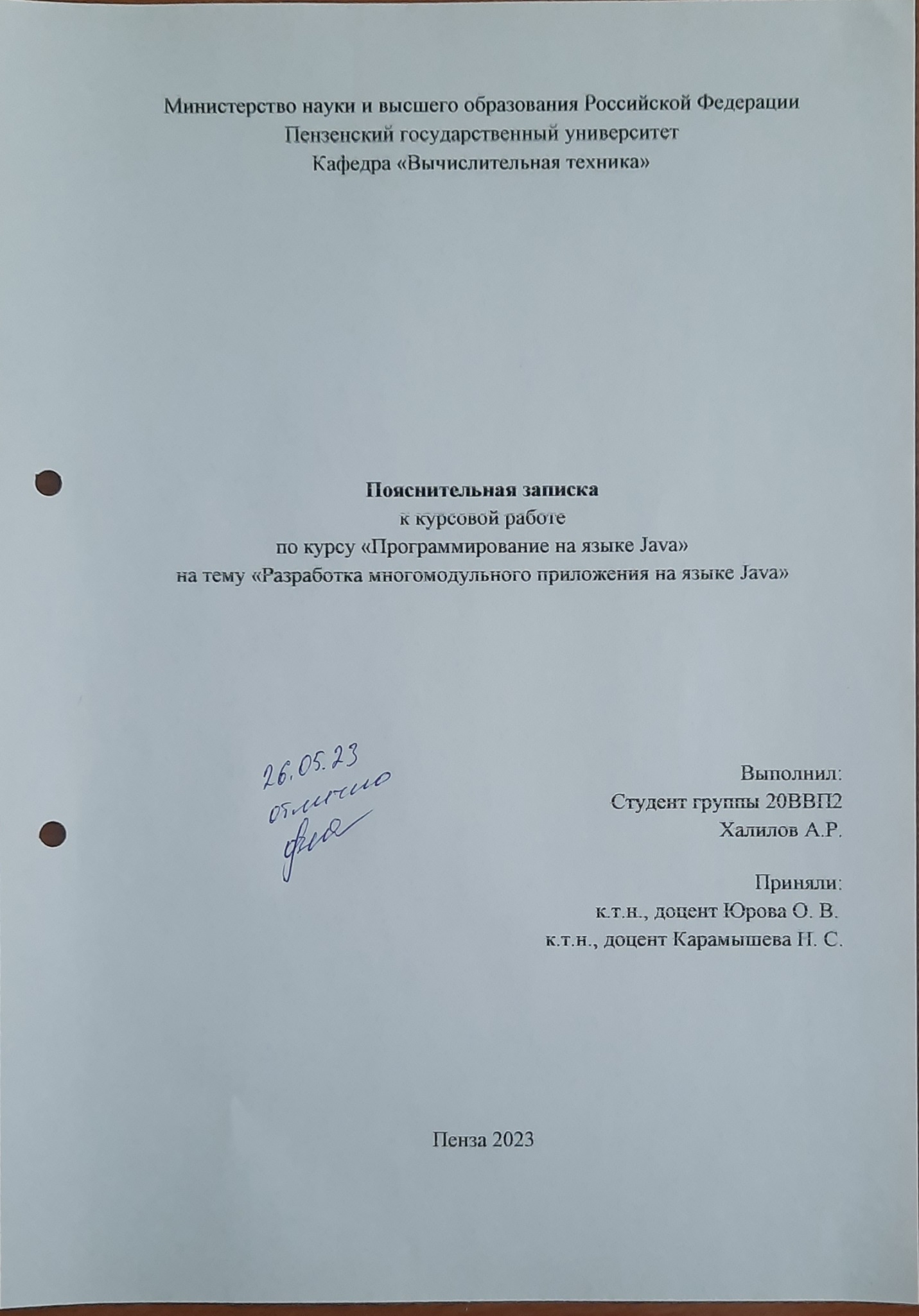
Студент группы 20ВВП2  
Халилов А.Р.

Приняли:

к.т.н., доцент Юрова О. В.

к.т.н., доцент Карамышева Н. С.

Пенза 2023



Содержание

[Введение 6](#_Toc135678358)

[1 Постановка задачи 7](#_Toc135678359)

[2 Выбор решения 8](#_Toc135678360)

[3 Описание программы 9](#_Toc135678361)

[3.1 Серверная часть 9](#_Toc135678362)

[3.2 Клиентская часть 9](#_Toc135678363)

[4 Описание способа организации пользовательского интерфейса 11](#_Toc135678364)

[5 Описание результатов работы программы 12](#_Toc135678365)

[5.1 Запуск программы и работа с плейлистами 12](#_Toc135678366)

[5.1 Выставление значений роторов 14](#_Toc135678367)

[5.2 Шифрование и расшифровка 15](#_Toc135678368)

[Заключение 17](#_Toc135678369)

[Список используемой литературы 18](#_Toc135678370)

[Приложение А. Исходный код 19](#_Toc135678371)

[Приложение А.1 – Исходный код Сервера 19](#_Toc135678372)

[Приложение А.2 – Исходный код Клиента 23](#_Toc135678373)

[Приложение В. UML-диаграммы приложения 31](#_Toc135678374)

[Приложение В.1 – UML-диаграмма вариантов использования 31](#_Toc135678375)

[Приложение В.2 – UML-диаграмма классов клиента 32](#_Toc135678376)

[Приложение В.3 – UML-диаграмма классов сервера 32](#_Toc135678377)

[Приложение В.4 – UML-диаграмма деятельности 33](#_Toc135678378)

[Приложение В.5 – UML-диаграмма развёртывания 34](#_Toc135678379)

[Приложение В.6 – UML-диаграмма последовательности 34](#_Toc135678380)

Введение

На сегодняшний день, Java является одним из наиболее широко распространенных и популярных языков программирования. Этот язык программирования является объектно-ориентированным и поддерживает такие принципы, как полиморфизм, наследование и статическая типизация. Благодаря объектно-ориентированному подходу, Java позволяет разрабатывать крупные, гибкие, масштабируемые и расширяемые приложения. Важно отметить, что приложения, написанные на Java, не зависят от конкретной платформы. Это обеспечивается наличием Java Virtual Machine (JVM) на каждой поддерживаемой платформе. Хотя JVM может отличаться на разных платформах, программы, написанные на Java, могут запускаться на любой из них.

Представленная в данном курсовом проекте программа является простым клиент-серверным приложением, которое позволяет пользователю шифровать и расшифровывать текст.

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 «Enigma» на языке Java в среде разработки NetBeans.

Программа должна работать так, чтобы пользователь мог печатать текст или загружать его из файла, выставлять значение роторов и сохранять результат в файл. 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

1. Выбор решения

Для реализации проекта используются:

* Объектно-ориентированный язык Java;
* Библиотека Swing;
* Пакеты, содержащие файлы ресурсов, пользовательский интерфейс;
* Java Stream API;
* IDE NetBeans 8.2;

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

Взаимодействие между клиентом и сервером должно происходить по принципу “Запрос-Ответ”. При этом для коротких и простых запросов используются более быстрый и простой протокол UDP, а для передаче с сервера больших объёмов информации, был использован протокол TCP, способный точно передавать большие объёмы данных.

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

* Поддержка ввода текста с клавиатуры;
* Выставление положений каждого ротора;
* Ввод текста с файла;
* Сохранение шифра в файл;
* Сохранение шифра в файл;

1. 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.

* 1. Серверная часть

Server.java – главный файл серверного процесса, котором находится точка входа в программу, функция Main.

После входа в программу сервер сканирует текст введенный пользователем.

При получении запроса сервер создаёт нить для его обработки. Метод run(), находящийся в нити, в котором происходит перестановка значений роторов:

* 1. Клиентская часть

Interface.java – главный файл пользовательского процесса, содержащий функции работы программы. Также реализует интерфейс главного меню программы.

Клиентская часть содержит следующие методы:

* private void FirsKeyActionPerformed – содержит функционал ввода значения первого ротора;
* private void SecondKeyActionPerformed – содержит функционал ввода значения второго ротора;
* private void ThirdKeyActionPerformed – содержит функционал ввода значения третьего ротора;
* private void StartBntActionPerformed – содержит функционал начала обработки введенного текста, проверка некорректного ввода значений роторов;
* void PlaySong(String SongName) – содержит функционал начала проигрывания новой композиции (отправка запроса на сервер, получение данных и первый запуск);
* private void OpenBntActionPerformed – содержит функционал ввода данных из файла;
* private void SaveBntActionPerformed – содержит функционал ввода данных в файл;

1. 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, запуска программы были использованы *jButton*.

Окна ввода значения трех роторов реализовано через *jFormattedTextField*

Большую часть окна приложения занимает поле ввода и вывода текста, для за его отображение отвечает компонент *jTextArea*.

1. 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 и работа с плейлистами

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

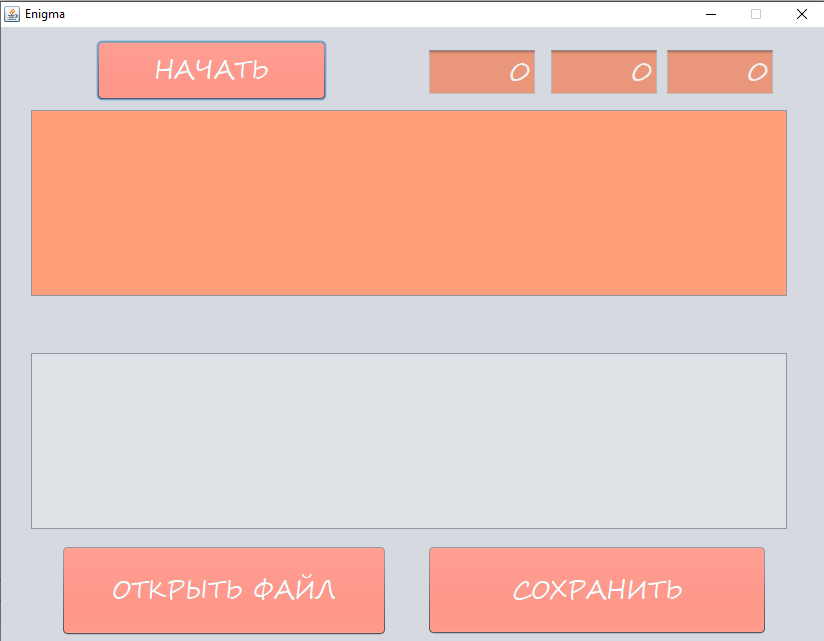


Рисунок – 1 Запуск приложения

При открыть или сохранить запись можно при помощи кнопок снизу. Загрузим готовый текст и посмотрим на результат (рис. 2).

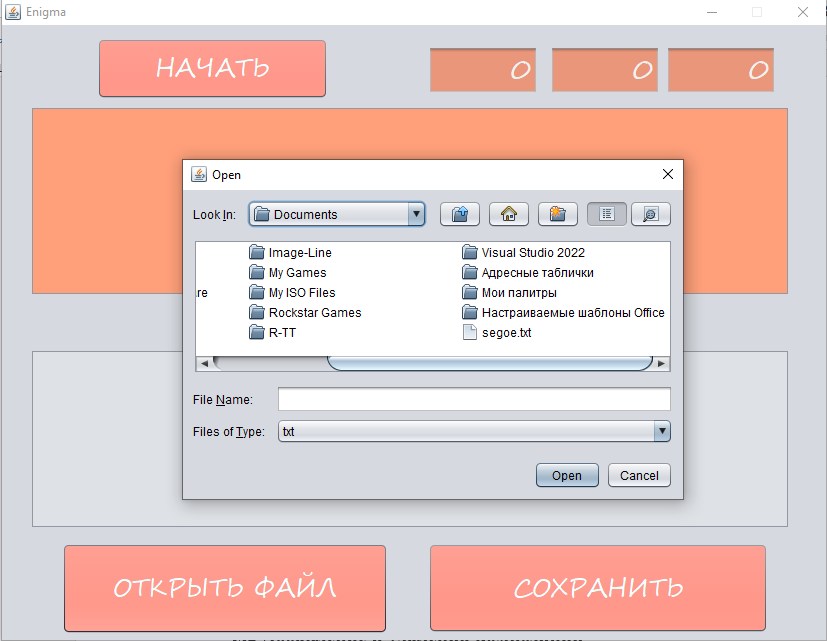
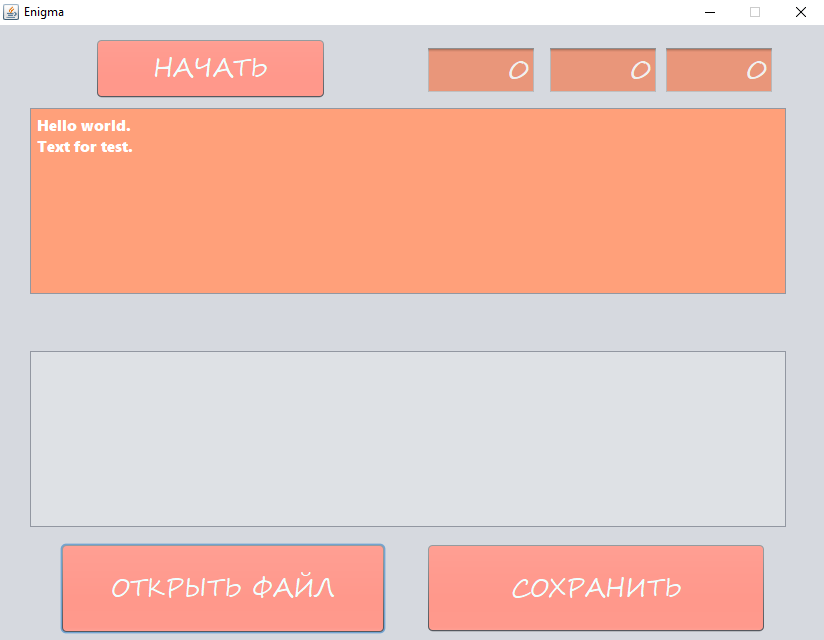
 

Рисунок – 2 Открытие файла

* 1. Выставление значений роторов

Для изменения положения роторов используются поля для ввода значения. Введенное значение проверяется, не является ли оно слишком большим или буквой.

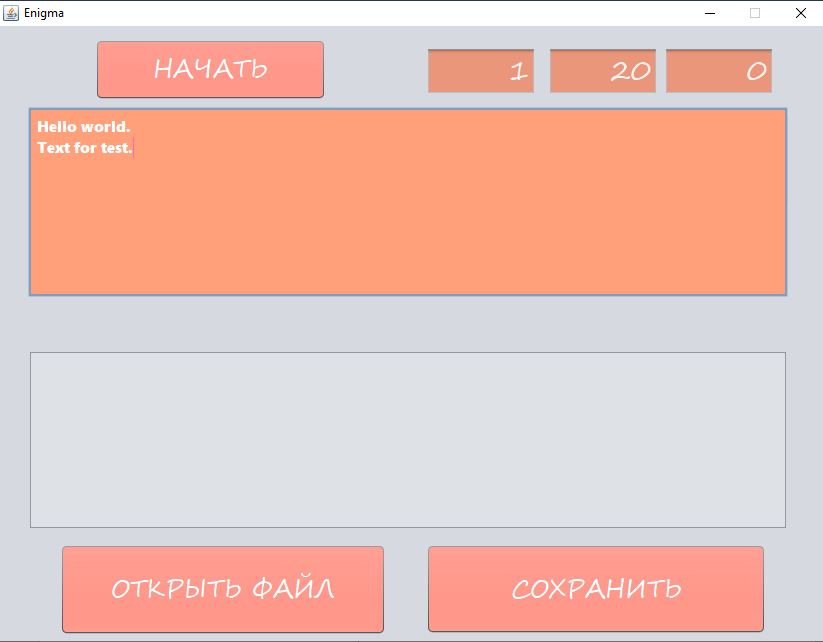


Рисунок – 3 Выставление значений роторов

* 1. Шифрование и расшифровка

Запустим программу с текстом (рис. 4).

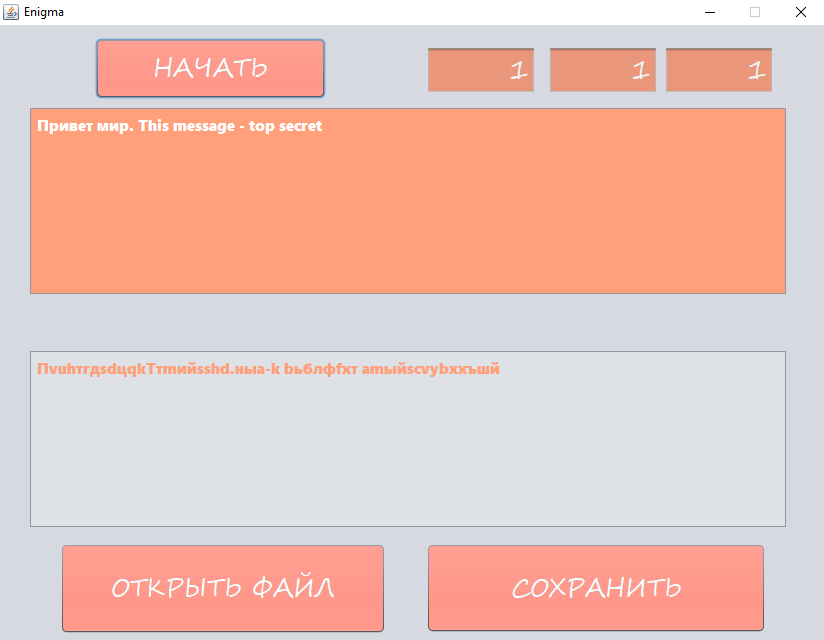


Рисунок – 3 Запуск проигрывателя

Далее мы можем увидеть результат шифрования. Сохраним его в файл, для дальнейшего открытия (рис. 4).

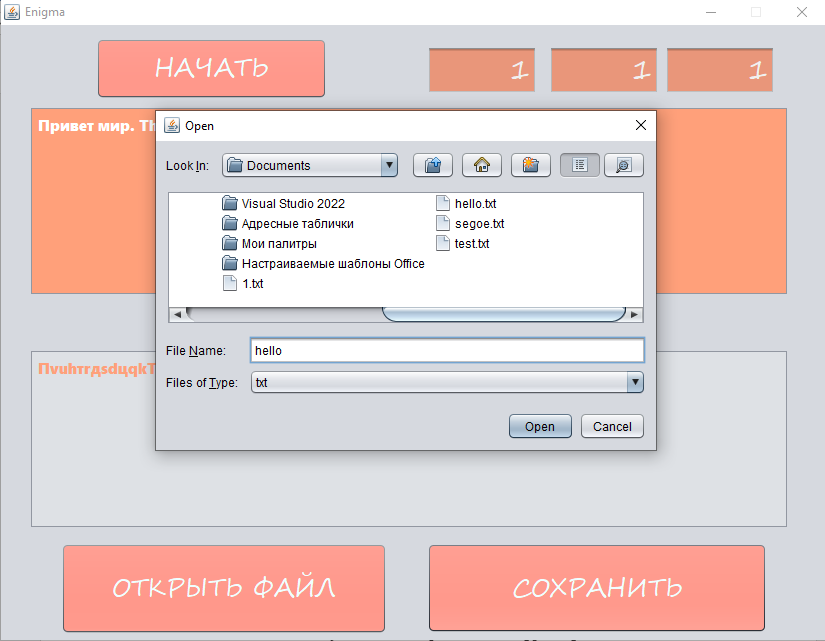


Рисунок – 4 Сохранение шифра в файл.

Откроем наш получившийся файл и прогоним его с тем же значением роторов(рис.5).

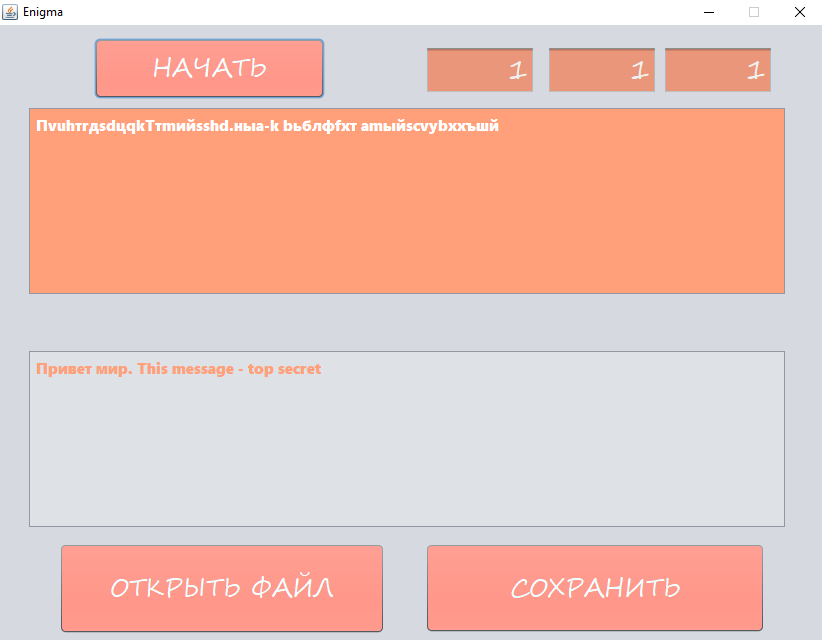


Рисунок 5 – дешифрация текста.

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 копию исторической шифрующей машинки, которая обладает основным функционалом, необходимым для подобных приложений в наше время.

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.

Приложение А. Исходный код

Приложение А.1 – Исходный код Сервера

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package server;

import com.sun.xml.internal.ws.util.StringUtils;

import java.io.IOException;

import java.lang.reflect.Array;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.net.SocketException;

import java.net.UnknownHostException;

import java.util.Arrays;

import java.util.logging.Level;

import java.util.logging.Logger;

/\*\*

\*

\* @author eltix

\*/

class JThread extends Thread

{

int Key1;

int Key2;

int Key3;

DatagramSocket socket;

InetAddress address;

String TextToEncode;

int[] rotor1 = { 0, 38 , 36 , 57 , 61 , 50 , 45 , 41 , 39 , 52 , 34 , 35 , 46 , 47 , 51 , 60 ,

59 , 48 , 56 , 42 , 55 , 49 , 37 , 40 , 58 , 43 , 33 , 44 , 53 , 62 , 54 , 32 , 31 ,

26 , 10 , 11 , 2 , 22 , 1 , 8 , 23 , 7 , 19 , 25 , 27 , 6 , 12 , 13 , 17 , 21 , 5 ,

14 , 9 , 28 , 30 , 20 , 18 , 3 , 24 , 16 , 15 , 4 , 29 };

int[] rotor2 = { 0, 59 , 35 , 40 , 60 , 32 , 56 , 38 , 42 , 50 , 62 , 47 , 36 , 55 , 57 , 52 ,

34 , 54 , 53 , 58 , 41 , 39 , 37 , 46 , 33 , 44 , 45 , 49 , 43 , 48 , 61 , 51 , 5 ,

24 , 16 , 2 , 12 , 22 , 7 , 21 , 3 , 20 , 8 , 28 , 25 , 26 , 23 , 11 , 29 , 27 , 9 ,

31 , 15 , 18 , 17 , 13 , 6 , 14 , 19 , 1 , 4 , 30 , 10 };

int[] rotor3 = { 0, 45 , 41 , 57 , 58 , 38 , 32 , 60 , 52 , 40 , 35 , 49 , 43 , 48 , 51 , 62 , 34 , 55 ,

61 , 59 , 50 , 37 , 53 , 36 , 39 , 42 , 56 , 47 , 54 , 33 , 46 , 44 , 6 , 29 , 16 , 10 , 23 , 21 , 5 ,

24 , 9 , 2 , 25 , 12 , 31 , 1 , 30 , 27 , 13 , 11 , 20 , 14 , 8 , 22 , 28 , 17 , 26 , 3 , 4 , 19 , 7 , 18 , 15 };

int[] reflector = { 0, 56 , 34 , 38 , 41 , 50 , 58 , 42 , 51 , 36 , 46 , 48 , 40 , 39 , 31 , 47 , 55 , 37 ,

59 , 33 , 52 , 60 , 32 , 44 , 54 , 53 , 49 , 45 , 43 , 35 , 57 , 14 , 22 , 19 , 2 , 29 , 9 , 17 , 3 ,

13 , 12 , 4 , 7 , 28 , 23 , 27 , 10 , 15 , 11 , 26 , 5 , 8 , 20 , 25 , 24 , 16 , 1 , 30 , 6 , 18 , 21 , 62 , 61 };

char[] abc = new char[]{ 0, 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r',

's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'а', 'б', 'в', 'г', 'д', 'е', 'Є', 'ж', 'з', 'и', 'й', 'к', 'л',

'м', 'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ' , 'ъ', 'ы' , 'ь' , 'э' , 'ю', '¤', ' ' , '.' ,',' };

JThread(String name, int \_a, int \_b, int \_c, DatagramSocket \_socket, InetAddress \_address,String text)

{

super(name);

Key1 = \_a;

Key2 = \_b;

Key3 = \_c;

TextToEncode = text;

socket = \_socket;

address = \_address;

}

public void run()

{

int razmr = 62;

String message = TextToEncode;

char[] TextToEncodeArray = TextToEncode.toCharArray();

char[] TextToOutput = new char[TextToEncodeArray.length];

Arrays.fill(TextToOutput,'~');

int marker;

for (int i = 0; i < TextToEncodeArray.length; i++)

{

if (Key1 < razmr-1)

Key1 = Key1 + 1;

else

{

Key1 = 0;

if (Key2 < razmr-1)

Key2 = Key2 + 1;

else

{

Key2 = 0;

if (Key3 < razmr-1)

Key3 = Key3 + 1;

else

{

Key3 = 0;

}

}

}

for (int j = 1; j != razmr && TextToOutput[i] == '~'; j++)

{

if (TextToEncodeArray[i] == abc[j])

{

int pmarker;

marker = j + Key1;

if (marker > razmr)

marker = marker - razmr;

marker = rotor1[marker]+Key2-Key1;

if (marker > razmr)

marker = marker - razmr;

if (marker < 1)

marker = marker + razmr;

marker = rotor2[marker] + Key3-Key2;

if (marker > razmr)

marker = marker - razmr;

if (marker < 1)

marker = marker + razmr;

marker = rotor3[marker] - Key3;

if (marker > razmr)

marker = marker - razmr;

if (marker < 1)

marker = marker + razmr;

marker = reflector[marker];

pmarker = marker;

if (pmarker + Key3 > razmr)

marker = marker - razmr;

if (pmarker + Key3 < 1)

marker = marker + razmr;

marker = rotor3[marker + Key3];

if (marker > razmr)

marker = marker - razmr;

pmarker = marker;

if (pmarker + Key2-Key3 > razmr)

marker = marker - razmr;

if (pmarker + Key2 - Key3 < 1)

marker = marker + razmr;

marker = rotor2[marker + Key2-Key3];

if (marker > razmr)

marker = marker - razmr;

pmarker = marker;

if (pmarker + Key1 - Key2 > razmr)

marker = marker - razmr;

if (pmarker + Key1 - Key2 < 1)

marker = marker + razmr;

marker = rotor1[marker + Key1-Key2];

if (marker > razmr)

marker = marker - razmr;

marker = marker - Key1;

if (marker < 0)

marker = marker + 62;

TextToOutput[i] = abc[marker];

}

if (j == (razmr-1) && TextToEncodeArray[i] != abc[j])

TextToOutput[i] = TextToEncodeArray[i];

}

}

String newMessage = String.valueOf(TextToOutput);

byte[] buf = newMessage.getBytes();

DatagramPacket packet = new DatagramPacket(buf, buf.length, address, 26);

try {

socket.send(packet);

} catch (IOException ex) {

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

}

}

}

public class Server {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) throws SocketException, UnknownHostException, IOException, InterruptedException {

DatagramSocket socket = new DatagramSocket(17);

DatagramSocket socketSend = new DatagramSocket();

int Key1,Key2,Key3;

InetAddress address = InetAddress.getByName("localhost");

while(true){

byte[] buffer = new byte[255];

DatagramPacket request = new DatagramPacket(buffer, buffer.length);

socket.receive(request);

if(request.getLength() != 0){

String Message = new String(request.getData(), 0, request.getLength());

String arr[] = Message.split(" ");

int MessageLendth = Integer.valueOf(arr[0]);

Key1 = Integer.valueOf(arr[1]);

Key2 = Integer.valueOf(arr[2]);

Key3 = Integer.valueOf(arr[3]);

byte[] TextBuffer = new byte[MessageLendth];

DatagramPacket EncodeReques = new DatagramPacket(TextBuffer, TextBuffer.length);

socket.receive(EncodeReques);

String TextToEncode = new String(EncodeReques.getData(), 0, EncodeReques.getLength());

JThread RunEncode = new JThread("RunEncode", Key1, Key2 , Key3 , socketSend, address, TextToEncode);

RunEncode.start();

}

}

}

}

Приложение А.2 – Исходный код Клиента

package enigma;

import java.io.BufferedOutputStream;

import java.io.BufferedWriter;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.io.ObjectOutputStream;

import java.io.StringReader;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.net.SocketException;

import java.net.UnknownHostException;

import static java.sql.JDBCType.NULL;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFileChooser;

import javax.swing.JOptionPane;

import javax.swing.filechooser.FileNameExtensionFilter;

import javax.swing.table.DefaultTableModel;

public class Interface extends javax.swing.JFrame {

DatagramSocket socet, socketIn;

InetAddress address;

public Interface() throws SocketException, UnknownHostException {

socet = new DatagramSocket();

address = InetAddress.getByName("localhost");

socketIn = new DatagramSocket(26);

initComponents();

}

/\*\*

\* 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() {

jScrollPane1 = new javax.swing.JScrollPane();

InText = new javax.swing.JTextArea();

jScrollPane3 = new javax.swing.JScrollPane();

OutText = new javax.swing.JTextArea();

SaveBnt = new javax.swing.JButton();

OpenBnt = new javax.swing.JButton();

StartBnt = new javax.swing.JButton();

FirsKey = new javax.swing.JFormattedTextField();

SecondKey = new javax.swing.JFormattedTextField();

ThirdKey = new javax.swing.JFormattedTextField();

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

setTitle("Enigma");

setBackground(new java.awt.Color(240, 128, 128));

setResizable(false);

setSize(new java.awt.Dimension(850, 650));

InText.setBackground(new java.awt.Color(255, 160, 122));

InText.setColumns(20);

InText.setFont(new java.awt.Font("Segoe UI Black", 0, 15)); // NOI18N

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

InText.setRows(5);

InText.setCaretColor(new java.awt.Color(255, 105, 180));

InText.setDoubleBuffered(true);

InText.setSelectionColor(new java.awt.Color(255, 160, 122));

jScrollPane1.setViewportView(InText);

OutText.setBackground(new java.awt.Color(255, 160, 122));

OutText.setColumns(20);

OutText.setFont(new java.awt.Font("Segoe UI Black", 0, 15)); // NOI18N

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

OutText.setRows(5);

OutText.setCaretColor(new java.awt.Color(255, 105, 180));

OutText.setDisabledTextColor(new java.awt.Color(255, 160, 122));

OutText.setEnabled(false);

jScrollPane3.setViewportView(OutText);

SaveBnt.setBackground(new java.awt.Color(240, 128, 114));

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

SaveBnt.setForeground(new java.awt.Color(240, 255, 255));

SaveBnt.setText("СОХРАНИТЬ");

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

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

SaveBntActionPerformed(evt);

}

});

OpenBnt.setBackground(new java.awt.Color(240, 128, 114));

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

OpenBnt.setForeground(new java.awt.Color(240, 255, 255));

OpenBnt.setText("ОТКРЫТЬ ФАЙЛ");

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

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

OpenBntActionPerformed(evt);

}

});

StartBnt.setBackground(new java.awt.Color(240, 128, 114));

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

StartBnt.setForeground(new java.awt.Color(240, 255, 255));

StartBnt.setText("НАЧАТЬ");

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

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

StartBntActionPerformed(evt);

}

});

FirsKey.setBackground(new java.awt.Color(233, 150, 122));

FirsKey.setForeground(new java.awt.Color(240, 240, 240));

FirsKey.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new javax.swing.text.NumberFormatter(new java.text.DecimalFormat("#0"))));

FirsKey.setHorizontalAlignment(javax.swing.JTextField.RIGHT);

FirsKey.setText("0");

FirsKey.setFont(new java.awt.Font("Segoe Script", 0, 28)); // NOI18N

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

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

FirsKeyActionPerformed(evt);

}

});

SecondKey.setBackground(new java.awt.Color(233, 150, 122));

SecondKey.setForeground(new java.awt.Color(240, 240, 240));

SecondKey.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new javax.swing.text.NumberFormatter(new java.text.DecimalFormat("#0"))));

SecondKey.setHorizontalAlignment(javax.swing.JTextField.RIGHT);

SecondKey.setText("0");

SecondKey.setFont(new java.awt.Font("Segoe Script", 0, 28)); // NOI18N

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

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

SecondKeyActionPerformed(evt);

}

});

ThirdKey.setBackground(new java.awt.Color(233, 150, 122));

ThirdKey.setForeground(new java.awt.Color(240, 240, 240));

ThirdKey.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new javax.swing.text.NumberFormatter(new java.text.DecimalFormat("#0"))));

ThirdKey.setHorizontalAlignment(javax.swing.JTextField.RIGHT);

ThirdKey.setText("0");

ThirdKey.setFont(new java.awt.Font("Segoe Script", 0, 28)); // NOI18N

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

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

ThirdKeyActionPerformed(evt);

}

});

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

.addGroup(layout.createSequentialGroup()

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

.addGroup(layout.createSequentialGroup()

.addGap(95, 95, 95)

.addComponent(StartBnt, javax.swing.GroupLayout.PREFERRED\_SIZE, 231, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(100, 100, 100)

.addComponent(FirsKey, javax.swing.GroupLayout.PREFERRED\_SIZE, 110, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(SecondKey, javax.swing.GroupLayout.PREFERRED\_SIZE, 110, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(ThirdKey, javax.swing.GroupLayout.PREFERRED\_SIZE, 110, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addGap(28, 28, 28)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 760, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addGap(28, 28, 28)

.addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED\_SIZE, 760, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addGap(60, 60, 60)

.addComponent(OpenBnt, javax.swing.GroupLayout.PREFERRED\_SIZE, 326, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(40, 40, 40)

.addComponent(SaveBnt, javax.swing.GroupLayout.PREFERRED\_SIZE, 340, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(36, 36, 36))

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

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

.addGroup(layout.createSequentialGroup()

.addGap(13, 13, 13)

.addComponent(StartBnt, javax.swing.GroupLayout.PREFERRED\_SIZE, 61, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(7, 7, 7))

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

.addContainerGap()

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

.addComponent(FirsKey, javax.swing.GroupLayout.PREFERRED\_SIZE, 48, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(SecondKey, javax.swing.GroupLayout.PREFERRED\_SIZE, 48, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(ThirdKey, javax.swing.GroupLayout.PREFERRED\_SIZE, 48, javax.swing.GroupLayout.PREFERRED\_SIZE))

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

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 190, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(53, 53, 53)

.addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED\_SIZE, 180, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(14, 14, 14)

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

.addComponent(OpenBnt, javax.swing.GroupLayout.PREFERRED\_SIZE, 91, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(SaveBnt, javax.swing.GroupLayout.PREFERRED\_SIZE, 90, javax.swing.GroupLayout.PREFERRED\_SIZE))

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

);

getAccessibleContext().setAccessibleName("ENIGMA JAVA EDITION");

pack();

}// </editor-fold>

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

OpenDialogBox odb = new OpenDialogBox();

String FileName = odb.DialogOpen("txt");

if(FileName != null){

try {

InText.read(new FileReader(FileName), NULL);

} catch (FileNotFoundException ex) {

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

} catch (IOException ex) {

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

}

}

}

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

OpenDialogBox odb = new OpenDialogBox();

String FileName = odb.DialogOpen("txt") + ".txt";

if(FileName != null){

try {

BufferedWriter out = new BufferedWriter(new FileWriter(FileName));

out.write(OutText.getText());

out.close();

JOptionPane.showMessageDialog(this, "Файл сохранён");

} catch (IOException ex) {

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

}

}

}

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

Long a = (Long)FirsKey.getValue();

if(a == null){

FirsKey.setValue(0);

}

if(a < 61){

JOptionPane.showMessageDialog(this, "Введено слишком большое значение ротора");

FirsKey.setValue(0);

}

}

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

Long a = (Long)SecondKey.getValue();

if(a == null){

SecondKey.setValue(0);

}

if(a < 61){

JOptionPane.showMessageDialog(this, "Введено слишком большое значение ротора");

SecondKey.setValue(0);

}

}

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

Long a = (Long)ThirdKey.getValue();

if(a == null){

ThirdKey.setValue(0);

}

if(a < 61){

JOptionPane.showMessageDialog(this, "Введено слишком большое значение ротора");

ThirdKey.setValue(0);

}

}

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

String TextToEncode = InText.getText();

byte[] buf;

buf = TextToEncode.getBytes();

Long a = (Long)FirsKey.getValue();

if(a == null)

a = Long.valueOf(0);

Long b = (Long)SecondKey.getValue();

if(b == null)

b = Long.valueOf(0);

Long c = (Long)ThirdKey.getValue();

if(c == null)

c = Long.valueOf(0);

if(a > 61 || b > 61 || c > 61 ){

JOptionPane.showMessageDialog(this, "Введено слишком большое значение ротора");

return;

}

int SizeOfText = buf.length;

String lendth = String.valueOf(SizeOfText) + " " + a + " " +

b + " " + c;

byte[] LendthBuf = lendth.getBytes();

DatagramPacket packet = new DatagramPacket(LendthBuf, LendthBuf.length, address, 17);

JThread CatchCodedText = new JThread("CatchCodedText", SizeOfText, socketIn);

CatchCodedText.start();

try {

socet.send(packet);

} catch (IOException ex) {

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

}

packet = new DatagramPacket(buf, buf.length, address, 17);

try {

socet.send(packet);

} catch (IOException ex) {

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

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

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

}

//</editor-fold>

/\* Create and display the form \*/

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

public void run() {

try {

new Interface().setVisible(true);

} catch (SocketException ex) {

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

} catch (UnknownHostException ex) {

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

}

}

});

}

class OpenDialogBox

{

String FileName;

public String DialogOpen(String NameFilter)

{

JFileChooser chooser = new JFileChooser();

FileNameExtensionFilter filter = new FileNameExtensionFilter(

"txt", NameFilter);

chooser.setFileFilter(filter);

int returnVal = chooser.showOpenDialog(null);

if(returnVal == JFileChooser.APPROVE\_OPTION)

{

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

}

return FileName;

}

}

class JThread extends Thread

{

int size;

DatagramSocket socket;

JThread(String name, int \_size, DatagramSocket soc)

{

super(name);

size = \_size;

socket = soc;

}

public void run()

{

for(int i = 0; i < size; i++)

{

byte[] buffer = new byte[size];

DatagramPacket request = new DatagramPacket(buffer, buffer.length);

try {

socket.receive(request);

String CodedText = new String(request.getData(), 0, request.getLength());

OutText.read(new StringReader(CodedText), null);

} catch (IOException ex) {

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

}

}

socket.close();

}

}

// Variables declaration - do not modify

private javax.swing.JFormattedTextField FirsKey;

private javax.swing.JTextArea InText;

private javax.swing.JButton OpenBnt;

private javax.swing.JTextArea OutText;

private javax.swing.JButton SaveBnt;

private javax.swing.JFormattedTextField SecondKey;

private javax.swing.JButton StartBnt;

private javax.swing.JFormattedTextField ThirdKey;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane3;

// End of variables declaration

}

Приложение В. UML-диаграммы приложения

Приложение В.1 – UML-диаграмма вариантов использования

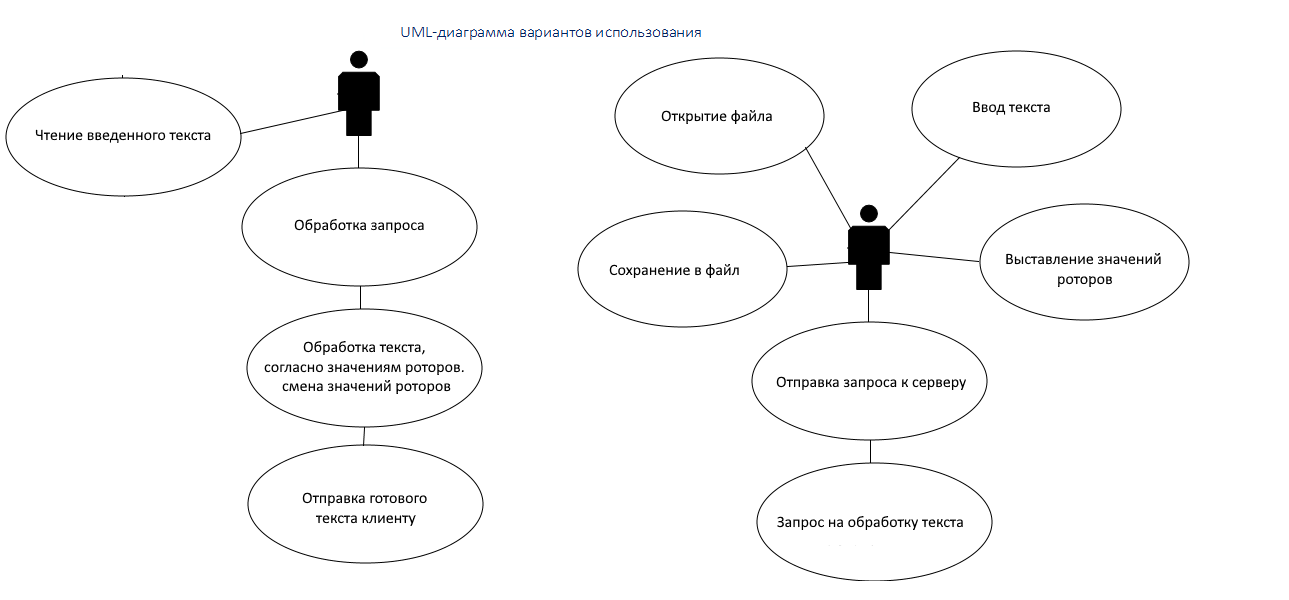


Рисунок 19 – UML-Диаграмма вариантов использования

Приложение В.2 – UML-диаграмма классов клиента



Рисунок 20 – UML-диаграмма классов клиента

Приложение В.3 – UML-диаграмма классов сервера



Рисунок 22 – UML-диаграмма классов сервера

Приложение В.4 – UML-диаграмма деятельности

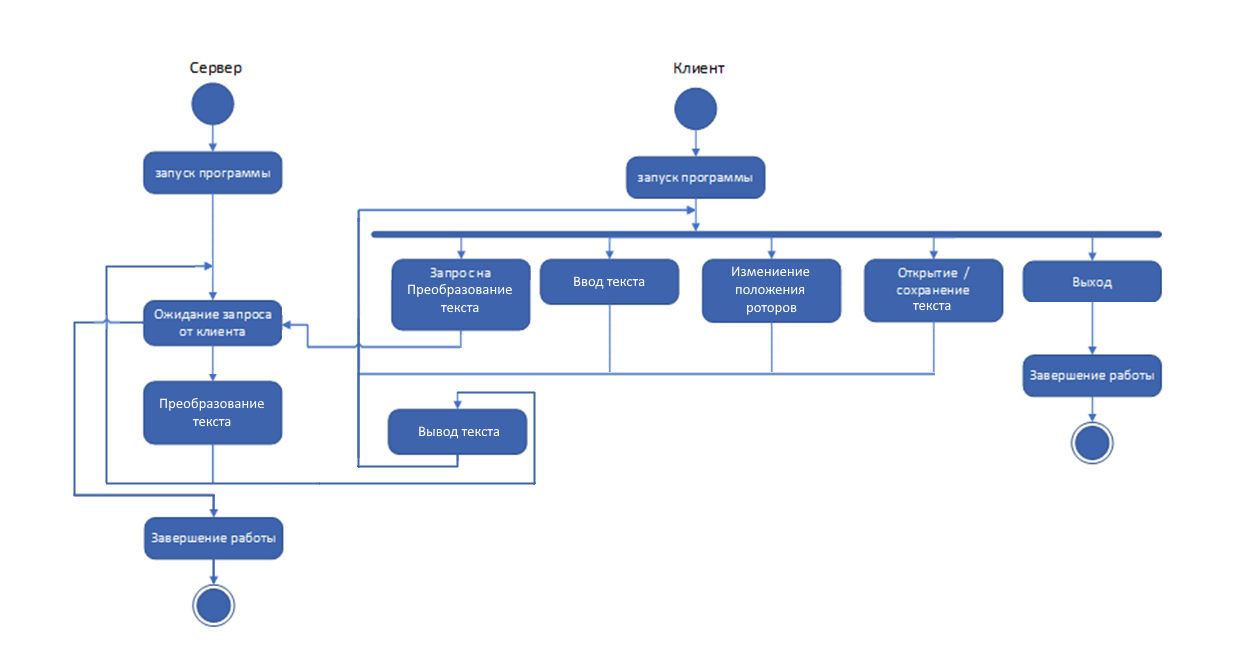


Рисунок 24 – UML-диаграмма деятельности

Приложение В.5 – UML-диаграмма развёртывания

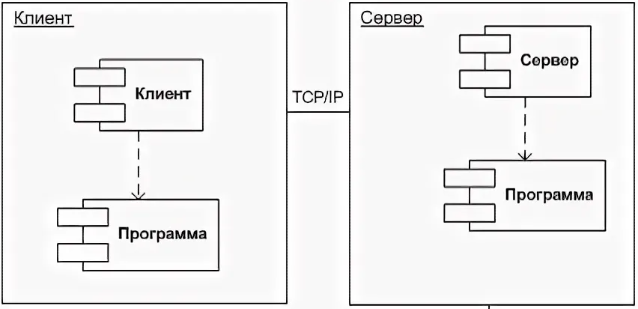


Рисунок 25 – UML-диаграмма развёртывания

Приложение В.6 – UML-диаграмма последовательности

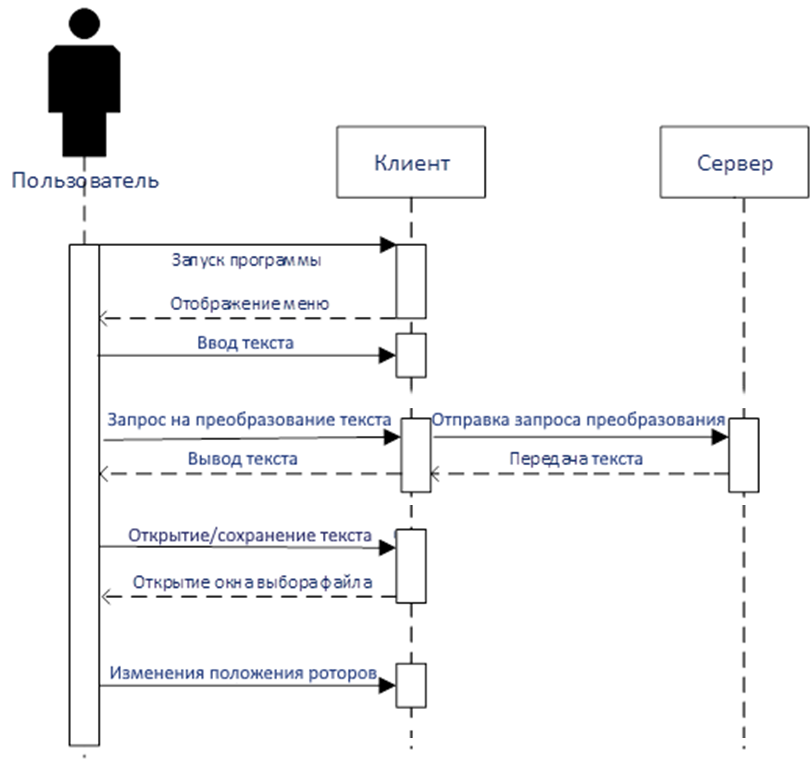


Рисунок 26 – UML-диаграмма последовательности