МИНИТЕРСТВО 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 19ВВ2

Гусев Д.О.

Приняла:

к.т.н., доцент Юрова О.В.

к.т.н., доцент Карамышева Н.С.

Пенза 2022

**Оглавление**

[Введение 3](#_Toc105024669)

[1. Постановка задачи 4](#_Toc105024670)

[2. Выбор решения 5](#_Toc105024671)

[3. Описание программы 6](#_Toc105024672)

[4. Экспериментальная часть 8](#_Toc105024673)

[5. Заключение 11](#_Toc105024674)

[Список используемой литературы. 12](#_Toc105024675)

[Приложение А. Исходный код серверной части. 13](#_Toc105024676)

[Приложение А.1 - MainForm.java 13](#_Toc105024677)

[Приложение Б. Исходный код клиентской части 18](#_Toc105024678)

[Приложение Б.1 – FiveClient.java 18](#_Toc105024679)

[Приложение Б.2 – Record.java 23](#_Toc105024680)

[Приложение В. UML-диаграммы приложения 24](#_Toc105024681)

[Приложение В.1 – UML-диаграмма вариантов использования 24](#_Toc105024682)

[Приложение В.2 – UML-диаграмма классов сервера 25](#_Toc105024683)

[Приложение В.3 – UML-диаграмма классов клиента 26](#_Toc105024684)

[Приложение В.4 – UML-диаграмма деятельности 27](#_Toc105024685)

[Приложение В.5 – UML-диаграмма развёртывания 28](#_Toc105024686)

[Приложение В.6 – UML-диаграмма последовательности 29](#_Toc105024687)

# Введение

Жизнь современного человека невозможно представить без ежедневного взаимодействия с компьютером. Сегодня люди с легкостью по всему миру могут обмениваться не только сообщениями, но и графикой и звуковой и другими видами информации.

Java – один из самых популярных языков программирования, с большим количеством достоинств:

1) Язык Java хорош для изучения объектно-ориентированного программирования.

2) Java имеет богатый API, и вы можете делать очень много в Java, например, работать с графикой, звуком и, что всего вероятнее, писать игры.

3) Java имеет сильную поддержку сообщества.

Такая простая игра, как «Пятнашки», отлично подходит для закрепления изученного в ходе курса материала.

# Постановка задачи

Разработать клиент-серверное приложение, реализующее игру «Пятнашки»

Функции сервера:

1) Обработка данных от пользователей.

2) Отправка результата в соответствии с принятыми данными

3) Своевременное окончание игровой сессии.

Функции клиента:

1. Реализация интерфейса, и его изменение.
2. Отправка данных серверу

Приложение должно обладать графическим интерфейсом и использовать следующие технологии:

1. Java Collections Framework
2. Механизм обработки исключительных ситуаций
3. Java Stream API
4. Java Multithreading
5. Сетевое взаимодействие.

ОС – Windows. Язык программирования: Java. Среда разработки NetBeans.

# 2. Выбор решения

Важной частью курсовой работы является передача сообщений от клиента к серверу. Для реализации этого используются TCP-сокеты.

Сокет— это один конец двустороннего канала связи между двумя программами, работающими в сети. Соединив вместе два сокета, можно передавать данные между разными процессами (локальными или удаленными). Для сокетов жизнь дескриптора можно разделить на три фазы: открыть (создать) сокет, получить из сокета или отправить сокету и в конце концов закрыть сокет.

Обычно приложение клиент-сервер, использующее сокеты, состоит из двух разных приложений - клиента, инициирующего соединение с целью (сервером), и сервера, ожидающего соединения от клиента.

Клиент соединяется с сервером с помощью библиотеки *java.net.Socket*.

Интерфейс пользователя реализован с помощью библиотеки Java Swing.

# 3. Описание программы

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

На других устройствах запускается клиентская часть, которая по нажатию определенной кнопки, пытается подключиться к серверу используя сокет.

**3.1 Серверная часть**

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

run() – метод, через который происходит взаимодействие с клиентами. Сервер получает данные от клиента. По окончанию игры он также получает данные о завершении игры и закрываются нити взаимодействия.

Вначале сервер ожидает подключения клиента через сокет. Он принимает сообщения «surname», «steps», и записывает их после окончания игры в файл.

**3.2 Клиентская часть.**

FiveClient.java – главный файл пользовательского процесса, содержащий функцию Main. Также реализует интерфейс.

Record.java – содержит интерфейс панели с рекордами.

run() – запускает нить ожидания запроса от сервера.

send(String msg) – отправляет данные на сервер.

get() – получает данные с сервера.

generate() – генерирует поле.

canBeSolved(int[] invariants) – булевая функция проверки на решаемость.

repaintField() – перерисовывание поля после нажатия.

checkWin() – проверка на выигрыш.

createMenu() – создание меню.

change(int num) – замена кнопок местами.

Клиент отправляют на сервер сообщения «win», «exit» или «rec». В зависимости от сообщения мы записываем в файл сообщения от клиента «surname» и «steps», выходим из программы или получаем рекорды из файла соответственно. Как только игра завершилась, появляется сообщение о выигрыше игрока, а на сервер отправляется сообщение о выходе. После этого игра закрывается.

# 4. Экспериментальная часть

Для начала запускаем сервер.

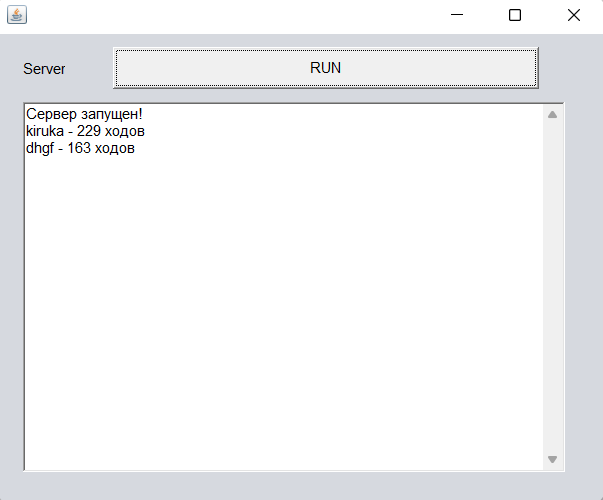


Рисунок – 1 Запуск сервера.

Далее запускаем клиента.

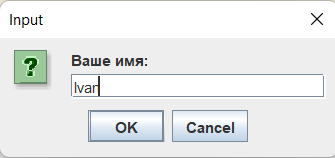


Рисунок – 2 Ввод имени.

После происходит успешное подключение к серверу Открывается игровое поле.

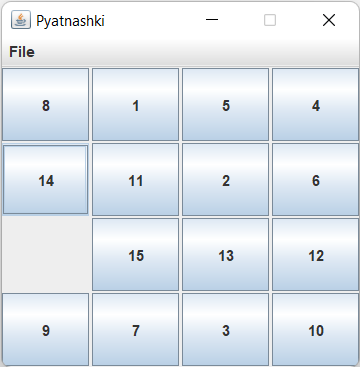


Рисунок – 3 Игровое поле.

Далее решаем головоломку. После её решения появиться окно с поздравлением и игра перезапуститься.

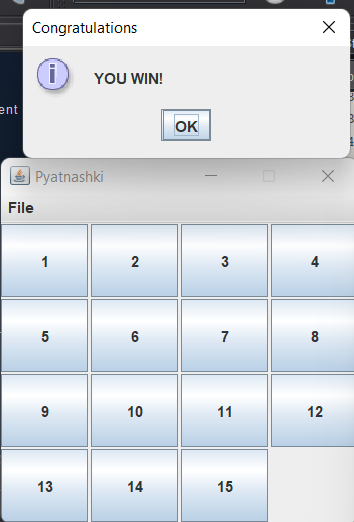


Рисунок – 4 Победа.

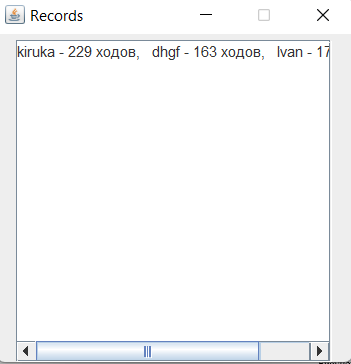


Рисунок – 5 Рекорды.

# 5. Заключение

При выполнении данной курсовой работы были получены навыки разработки программ на языке высокого уровня Java. Были освоены навыки разработки графического интерфейса с помощью библиотеки Swing. Изучены возможности среды разработки NetBeans. Освоены навыки разработки клиент-серверного приложения.

В рамках данной работы было написано приложение «Пятнашки».

# Список используемой литературы.

1. Берд, Барри Java для чайников / Барри Берд. - М.: Диалектика / Вильямс, 2013. - 521 c.

2. Гурвиц Г. Разработка реального приложения в среде клиент-сервер –

ДВГУПС, 2005, 206 с.

3. Дубаков А.А. Сетевое программирование: учебное пособие / А.А. Дубаков – СП: НИУ ИТМО, 2013. – 248 с.

4. Мейер Б. Объектно-ориентированное программирование и программная инженерия / Мейер Б.— Электрон. текстовые данные.— М.: Интернет Университет Информационных Технологий (ИНТУИТ), 2016.— 285с.

# Приложение А. Исходный код серверной части.

# Приложение А.1 - MainForm.java

package com.mycompany.labserver;

import java.io.\*;

import static java.lang.Thread.sleep;

import java.net.ServerSocket;

import java.net.Socket;

import java.util.logging.Level;

import java.util.logging.Logger;

public class Mainform extends javax.swing.JFrame

{

public Mainform()

{

initComponents();

}

private static Socket clientSocket; //сокет для общения

private static ServerSocket server; // серверсокет

private static BufferedReader in; // поток чтения из сокета

private static BufferedWriter out; // поток записи в сокет

String command;

/\*\*

\* 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">//GEN-BEGIN:initComponents

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

RunButton = new java.awt.Button();

textArea1 = new java.awt.TextArea();

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

jLabel1.setText("Server");

RunButton.setLabel("RUN");

RunButton.setName("RunButton"); // NOI18N

RunButton.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

RunButtonMouseClicked(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()

.addGap(20, 20, 20)

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

.addGroup(layout.createSequentialGroup()

.addComponent(textArea1, javax.swing.GroupLayout.PREFERRED\_SIZE, 434, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(31, Short.MAX\_VALUE))

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel1)

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

.addComponent(RunButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 341, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(52, 52, 52))))

);

layout.setVerticalGroup(

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

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

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

.addGroup(layout.createSequentialGroup()

.addGap(19, 19, 19)

.addComponent(jLabel1))

.addGroup(layout.createSequentialGroup()

.addGap(11, 11, 11)

.addComponent(RunButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 34, javax.swing.GroupLayout.PREFERRED\_SIZE)))

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

.addComponent(textArea1, javax.swing.GroupLayout.PREFERRED\_SIZE, 296, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(24, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>//GEN-END:initComponents

Thread thread = new Thread() {

@Override

public void run() {

try

{

server = new ServerSocket(4004); // серверсокет прослушивает порт 4004

try {

sleep(1);

} catch (InterruptedException ex) {

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

}

textArea1.append("Сервер запущен!\n");

try {

File file = new File("records.txt");

FileReader fr = new FileReader(file);

BufferedReader reader = new BufferedReader(fr);

String line = reader.readLine();

while (line != null) {

textArea1.append(line + "\n");

line = reader.readLine();

}

} catch (FileNotFoundException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

clientSocket = server.accept();

in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

out = new BufferedWriter(new OutputStreamWriter(clientSocket.getOutputStream()));

while (true){

get();

if("win".equals(command)) {

String surname = in.readLine();

String steps = in.readLine();

textArea1.append(surname + " - " + steps + " ходов\n");

try(FileWriter writer = new FileWriter("records.txt", true))

{

writer.append(surname + " - " + steps + " ходов\n");

writer.flush();

}

catch(IOException ex){

System.out.println(ex.getMessage());

}

}

if ("rec".equals(command)){

String text = "";

try {

File file = new File("records.txt");

FileReader fr = new FileReader(file);

BufferedReader reader = new BufferedReader(fr);

String line = reader.readLine();

while (line != null) {

text = text + line + ", ";

line = reader.readLine();

}

} catch (FileNotFoundException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

send(text);

}

if("exit".equals(command)){

textArea1.append("Сервер закрыт!");

System.out.println("Сервер закрыт!");

clientSocket.close();

in.close();

out.close();

break;

}

}

server.close();

}

catch (IOException e)

{

System.err.println(e);

}

}

};

private void RunButtonMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_RunButtonMouseClicked

thread.start();

}//GEN-LAST:event\_RunButtonMouseClicked

/\*\*

\* @param args the command line arguments

\*/

private void send(String msg) {

try {

out.write(msg + "\n");

out.flush();

} catch (IOException ignored) {}

}

private void get() {

try {

command = in.readLine();

} catch (IOException ignored) {}

}

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()

{

public void run()

{

new Mainform().setVisible(true);

}

});

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private java.awt.Button RunButton;

private javax.swing.JLabel jLabel1;

private java.awt.TextArea textArea1;

// End of variables declaration//GEN-END:variables

}

# Приложение Б. Исходный код клиентской части

## Приложение Б.1 – FiveClient.java

/\*

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

\* Click nbfs://nbhost/SystemFileSystem/Templates/Project/Maven2/JavaApp/src/main/java/${packagePath}/${mainClassName}.java to edit this template

\*/

package com.mycompany.fiveclient;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.OutputStreamWriter;

import java.util.Random;

import java.net.Socket;

import java.util.logging.Level;

import java.util.logging.Logger;

public class FiveClient extends JFrame {

private JPanel panel = new JPanel(new GridLayout(4, 4, 2, 2));

private int[][] numbers = new int[4][4];

private static Socket clientSocket; //сокет для общения

private static BufferedReader in; // поток чтения из сокета

private static BufferedWriter out; // поток записи в сокет

String surname;

String record;

int steps = 0;

Runnable thread = new Runnable(){

@Override

public void run() {

try {

get();

Thread.sleep(100);

} catch (InterruptedException ex) {

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

}

}

};

public FiveClient() {

super("Pyatnashki");

surname = JOptionPane.showInputDialog("Ваше имя: ");

setBounds(200, 200, 300, 300);

setResizable(false);

setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

createMenu();

Container container = getContentPane();

panel.setDoubleBuffered(true);

container.add(panel);

generate();

repaintField();

}

public void generate() {

Random generator = new Random();

int[] invariants = new int[16];

steps = 0;

do {

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 4; j++) {

numbers[i][j] = 0;

invariants[i \* 4 + j] = 0;

}

}

for (int i = 1; i < 16; i++) {

int k, l;

do {

k = generator.nextInt(4);

l = generator.nextInt(4);

}

while (numbers[k][l] != 0);

numbers[k][l] = i;

invariants[k \* 4 + l] = i;

}

}

while (!canBeSolved(invariants));

}

private boolean canBeSolved(int[] invariants) {

int sum = 0;

for (int i = 0; i < 16; i++) {

if (invariants[i] == 0) {

sum += i / 4;

continue;

}

for (int j = i + 1; j < 16; j++) {

if (invariants[j] < invariants[i])

sum ++;

}

}

System.out.println(sum % 2 == 0);

return sum % 2 == 0;

}

public void repaintField() {

panel.removeAll();

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 4; j++) {

JButton button = new JButton(Integer.toString(numbers[i][j]));

button.setFocusable(false);

panel.add(button);

if (numbers[i][j] == 0) {

button.setVisible(false);

} else

button.addActionListener(new ClickListener());

}

}

panel.validate();

panel.repaint();

}

public boolean checkWin() {

boolean status = true;

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 4; j++) {

if (i == 3 && j > 2)

break;

if (numbers[i][j] != i \* 4 + j + 1) {

status = false;

}

}

}

return status;

}

private void createMenu() {

JMenuBar menu = new JMenuBar();

JMenu fileMenu = new JMenu("File");

for (String fileItem : new String [] { "New", "Records", "Exit"}) {

JMenuItem item = new JMenuItem(fileItem);

item.setActionCommand(fileItem.toLowerCase());

item.addActionListener(new NewMenuListener());

fileMenu.add(item);

}

menu.add(fileMenu);

setJMenuBar(menu);

}

private class NewMenuListener implements ActionListener {

public void actionPerformed(ActionEvent e) {

String command = e.getActionCommand();

if ("exit".equals(command)) {

try {

send("exit");

clientSocket.close();

in.close();

out.close();

System.exit(0);

} catch (IOException ex) {

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

}

}

if ("new".equals(command)) {

generate();

repaintField();

}

if ("records".equals(command)){

send("rec");

Thread getRec = new Thread(thread);

getRec.start();

try {

getRec.join();

} catch (InterruptedException ex) {

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

}

JFrame rec = new Records(record);

rec.setVisible(true);

}

}

}

private class ClickListener implements ActionListener {

public void actionPerformed(ActionEvent e) {

JButton button = (JButton) e.getSource();

button.setVisible(false);

String name = button.getText();

change(Integer.parseInt(name));

}

}

public void change(int num) {

int i = 0, j = 0;

for (int k = 0; k < 4; k++) {

for (int l = 0; l < 4; l++) {

if (numbers[k][l] == num) {

i = k;

j = l;

}

}

}

if (i > 0) {

if (numbers[i - 1][j] == 0) {

numbers[i - 1][j] = num;

numbers[i][j] = 0;

}

}

if (i < 3) {

if (numbers[i + 1][j] == 0) {

numbers[i + 1][j] = num;

numbers[i][j] = 0;

}

}

if (j > 0) {

if (numbers[i][j - 1] == 0) {

numbers[i][j - 1] = num;

numbers[i][j] = 0;

}

}

if (j < 3) {

if (numbers[i][j + 1] == 0) {

numbers[i][j + 1] = num;

numbers[i][j] = 0;

}

}

steps++;

repaintField();

if (checkWin()) {

send("win");

send(surname);

send(Integer.toString(steps));

JOptionPane.showMessageDialog(null, "YOU WIN!", "Congratulations", JOptionPane.INFORMATION\_MESSAGE);

generate();

repaintField();

}

}

public static void main(String[] args) {

try {

JFrame app = new FiveClient();

app.setVisible(true);

clientSocket = new Socket("localhost", 4004); // этой строкой мы запрашиваем

in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

out = new BufferedWriter(new OutputStreamWriter(clientSocket.getOutputStream()));

} catch (IOException ex) {

System.err.println(ex);

}

}

private void send(String msg) {

try {

out.write(msg + "\n");

out.flush();

} catch (IOException ignored) {}

}

private void get() {

try {

record = in.readLine();

} catch (IOException ignored) {}

}

}

# Приложение Б.2 – Record.java

package com.mycompany.fiveclient;

import java.awt.Container;

import java.awt.GridLayout;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTextArea;

/\*\*

\*

\* @author Kiruka

\*/

public class Records extends JFrame {

private JPanel panel = new JPanel();

String text;

public Records(String text) {

super("Records");

setBounds(200, 200, 300, 300);

setResizable(false);

setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);

Container container = getContentPane();

panel.setDoubleBuffered(true);

JTextArea area = new JTextArea(text, 15, 25);

area.setEditable(false);

container.add(panel);

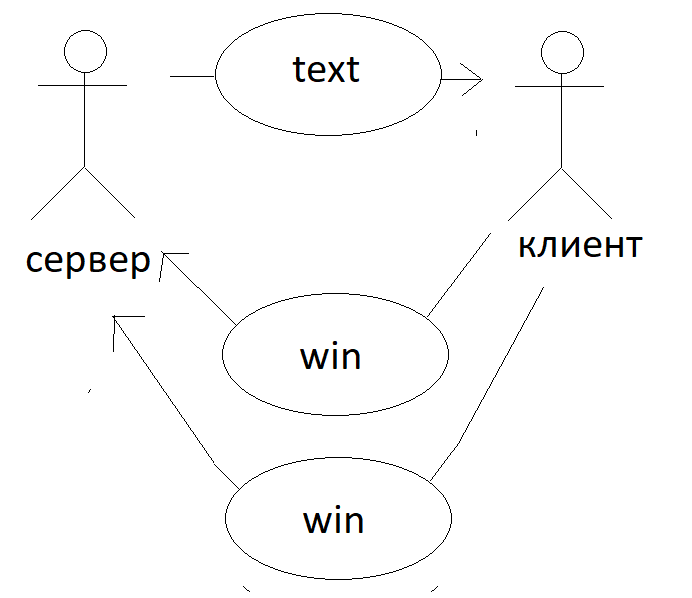
panel.add(new JScrollPane(area));

}

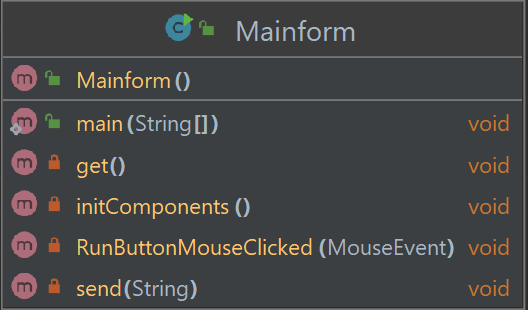
}

# Приложение В. UML-диаграммы приложения

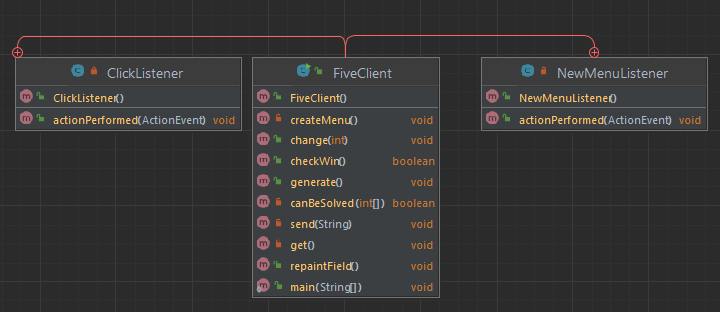
# Приложение В.1 – UML-диаграмма вариантов использования



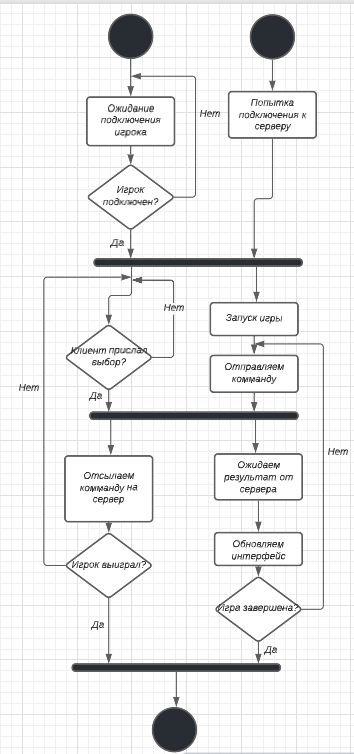
## Приложение В.2 – UML-диаграмма классов сервера



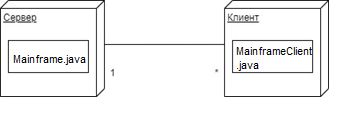
## Приложение В.3 – UML-диаграмма классов клиента



# Приложение В.4 – UML-диаграмма деятельности



# Приложение В.5 – UML-диаграмма развёртывания



MainForm.java

Record.java

FiveClient.java

# Приложение В.6 – UML-диаграмма последовательности

