

Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

по курсу «Разработка кроссплатформенных приложений»

на тему «Разработка многомодульного приложения на языке Java.

Игра «Сапер»

Выполнил:

студент группы 17BB1

Дмитров Н.А.

Принял:

к.т.н., доцент Юрова О.В.

Пенза 2020

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет Вычислительной техники

Кафедра "Вычислительная техника"

"УТВЕРЖДАЮ"

Зав. кафедрой ВТ

профессор М.А. Митрохин

« ____ » _____ 20 __ г

ЗАДАНИЕ

на курсовое проектирование по курсу

Разработка кроссплатформенных приложений
Студенту Миширову Никите Андреевичу Группа 17ВВ1
Тема проекта Разработка многомодульного приложения на языке Java. Игра "Сапёр"

Исходные данные (технические требования) на проектирование

Разработать многомодульное приложение:
Игра "Сапёр".
Язык программирования: Java
Среда разработки: NetBeans
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

Объем работы по курсу

1. Расчетная часть

2. Графическая часть

Диаграммы языка UML
1) диаграмма классов; 2) диаграмма деятельности
3) диаграмма вариантов использования;
4) диаграмма развертывания; 5) диаграмма последовательности

3. Экспериментальная часть

Тестирование многомодульного приложения

Срок выполнения проекта по разделам

1 в соответствии с графиком

2 _____
3 _____
4 _____
5 _____
6 _____
7 _____
8 _____

Дата выдачи задания " _____ "

Дата защиты проекта " _____ "

Руководитель Юрова Ольга Викторовна

Задание получил " 18 " феврале 2020г.

Студент Филиппов И.А. Рис

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	4
ВВЕДЕНИЕ.....	6
1 Постановка задачи.....	9
2 Теоретическая часть задания	10
2.1 Система программирования.....	10
2.2 Правила игры.....	10
3 Описание алгоритма работы	12
4 Описание программы.....	14
4.1 Интерфейс клиентской части.....	14
4.2 Интерфейс окна «Настройки».....	15
4.3 Интерфейс окна «Рейтинг игроков»	16
4.4 Интерфейс графической оболочки сервера.....	17
ЗАКЛЮЧЕНИЕ	18
СПИСОК ЛИТЕРАТУРЫ	19
Приложение А. Листинг программы клиента.....	20
Приложение А.1. Файл «Window.java».....	20
Приложение А.2. Файл «Cell.java».....	34
Приложение А.3. Файл «Field.java»	34
Приложение А.4. Файл «FieldComponent.java»	36
Приложение А.5. Файл «Result.java»	42
Приложение А.6. Файл «ListResult.java»	43
Приложение А.7. Файл «CanNotOpenException.java».....	44
Приложение А.8. Файл «ServerThread.java».....	44
Приложение Б. Листинг программы сервера.....	46
Приложение Б.1. Файл «Window.java»	46
Приложение Б.2. Файл «Result.java».....	48
Приложение Б.3. Файл «ListResult.java»	49

Приложение Б.4. Файл «ServerThread.java»	50
Приложение Б.5. Файл «ClientHandlerThead.java»	51
Приложение В. UML-диаграммы	53
Приложение В.1. UML-диаграмма вариантов использования	53
Приложение В.2. UML- диаграмма последовательности	54
Приложение В.3. UML-диаграмма классов.....	55
Приложение В.4. UML-диаграмма развертывания.....	56

ВВЕДЕНИЕ

История создания языка Java начинается в июне 1991 года, когда Джеймс Гослинг создал проект для использования в одном из своих многочисленных сет-топ проектов. Язык, который рос вне офиса Гослинга, как дуб, **Oak** - первоначальное название Java до 1995 года, после в дальнейшем история Java продолжалась под именем **Green**, а позже был переименован как Java.

Но официальной датой создания языка Java считается 23 мая 1995 года, после выпуска компанией Sun первой реализации Java 1.0. Она гарантировала «**Напиши один раз, запускай везде**», обеспечивая недорогой стоимостью на популярных платформах.

13 ноября 2006 года, Sun выпустила большую часть как свободное и открытое программное обеспечение в соответствии с условиями GNU General Public License (GPL).

После 8 мая 2007 года судьба Java сложилась иначе. Компания завершила процесс, делая все чтобы исходный код был бесплатным и открытым, кроме небольшой части кода, на который компания не имела авторских прав.

Преимущества языка Java: краткий обзор

Объектно-ориентированный язык: в Java все является объектом. Дополнение может быть легко расширено, так как он основан на объектной модели.

Платформонезависимый: в отличие от многих других языков, включая C и C++, Java, когда был создан, он не компилировался в платформе конкретной машины, а в независимом от платформы байт-коде. Этот байт код распространяется через интернет и интерпретируется в Java Virtual Machine (JVM), на которой он в настоящее время работает.

Простой: процессы изучения и введение в язык программирования Java остаются простыми. Если Вы понимаете основные концепции объектно-ориентированного программирования, то он будет прост для Вас в освоении.

Безопасный: методы проверки подлинности основаны на шифровании с открытым ключом.

Архитектурно-нейтральным: компилятор генерирует архитектурно-нейтральные объекты формата файла, что делает скомпилированный код исполняемым на многих процессорах, с наличием системе Java Runtime.

Портативный: архитектурно-нейтральный и не имеющий зависимости от реализации аспектов спецификаций — все это делает Java портативным. Компилятор в Java написан на ANSI C с чистой переносимостью, который является подмножеством POSIX.

Прочный: прилагает усилия, чтобы устранить ошибки в различных ситуациях, делая упор в основном на время компиляции, проверку ошибок и проверку во время выполнения.

Многопоточный: функции многопоточности, можно писать программы, которые могут выполнять множество задач одновременно. Введение в язык Java этой конструктивной особенности позволяет разработчикам создавать отлаженные интерактивные приложения.

Интерпретированный: Java байт-код переводится на лету в машинные инструкции и нигде не сохраняется. Делая процесс более быстрым и аналитическим, поскольку связывание происходит как дополнительное с небольшим весом процесса.

Высокопроизводительный: введение Just-In-Time компилятора, позволило получить высокую производительность.

Распространенный: предназначен для распределенной среды интернета.

Динамический: программирование на Java считается более динамичным, чем на С или С++, так как он предназначен для адаптации к меняющимся условиям. Программы могут выполнять обширное количество во время обработки информации, которая может быть использована для проверки и разрешения доступа к объектам на время выполнения.

1 Постановка задачи

Разработать многомодульное приложение «Игра «Сапер». Язык программирования Java, среда разработки NetBeans. Приложение должно обладать графическим интерфейсом и использовать следующие технологии:

1. Java Collection Framework;
2. Механики обработки исключительных ситуаций;
3. Java Stream API;
4. Java Multithreading;
5. Сетевое взаимодействие.

2 Теоретическая часть задания

2.1 Система программирования

NetBeans IDE - бесплатная интегрированная среда разработки с открытым исходным кодом для разработчиков программного обеспечения. Среда предоставляет все средства, необходимые для создания профессиональных десктоп приложений, корпоративных, мобильных и веб-приложений на платформе Java, а также C/C++, PHP, JavaScript, Groovy и Ruby.

Основные характеристики NetBeans IDE:

- Рабочая область среды IDE является полностью настраиваемой - существует возможность пользовательской настройки действий, выполняемых с помощью панели, назначения "горячих" клавиш и т.д.
- IDE имеет в своем составе расширенный многоязыковой редактор для различных языков программирования - Java, C/C++, Ruby, Groovy, PHP, JavaScript, CSS, XML, HTML, RHTML, JSP, документацию Javadoc. Существует возможность расширения функций редактора с целью поддержки любого другого языка.
- Редактор NetBeans делает отступы строк, проверяет соответствие скобок и слов, подсвечивает синтаксис исходного кода.

2.2 Правила игры

«Сапер» – это несложная игра, развивающая внимание, память и логику. Цель – открыть все ячейки на поле так, чтобы не “взорвать” ни одну мину. Для поиска взрывоопасных ячеек использовать подсказки:

- 1 Числа в ячейках показывают количество мин, скрытых в восьми находящихся рядом ячейках;
- 2 Если возле открытой ячейки есть пустые клеточки, они откроются автоматически;
- 3 Если вы понимаете, что в какой-то клеточке скрывается бомба, пометьте ее флажком;

4 Когда вы откроете ячейку с миной, игра заканчивается проигрышем; Для победы в игре необходимо отметить все «бомбы» флажками.

Партия начинается после первой открытой ячейки. С этого момента запускается таймер, который учитывает время на решение головоломки.

Значение цифр в клетках

Числовые значения на клетках указывают на количество бомб в соседних.

С помощью этих подсказок можно определить, где расположены мины:

- число 1 указывает, что из клеток сверху, слева, снизу, справа и по диагонали от открытой лишь одна содержит ловушку.
- вокруг ячеек с числом 2 находится по 2 бомбы;
- с числом 3 – по 3.

Если цифра в открытой клетке отсутствует, вокруг располагаются лишь пустые.

3 Описание алгоритма работы

3.1 Описание клиент-серверной части

Разработанная система команд состоит из клиентской и серверной частей. Для того что бы начать игровую партию не обязательно иметь подключение к серверу. Сервер выполняет функцию сбора и систематизации данных о достижениях игроков.

Для передачи данных использовался протокол TCP/IP

3.2 Описание интерфейса игры

При запуске приложения открывается окно с меню и игровым полем 9 на 9 клеток, на данном поле расположено 15 мин, которые случайным образом распределяются по полю.

Сверху окна находятся 3 кнопки. «Начать заново», «Настройки», «Рейтинг игроков» соответственно в порядке слева на право.

«Начать заново» - перезапускает игру, с теми же настройками что и были установлены до этого. После запуска порядок мин меняется.

«Настройки» - открывает меню, где можно изменить размер игрового поля, а также количество мин. Минимальный размер поля 9 на 9, минимальное количество мин 1. Так же в данном меню существует кнопка «Соревнование», которая использует предварительно заранее установленные настройки. С таким набором настроек, время за которое была разгадана головоломка будет отправляться на сервер для составления рейтинга игроков.

«Рейтинг игроков» - таблица, которая сортируется по количеству времени, затраченному на решение головоломки в порядке возрастания. Все игроки решают головоломки, которые имеют одинаковый размер поля количество мин для того, чтобы результат был занесен в таблицу. Данные клиент берет с сервера при запуске и каждый раз, когда отправляет данные на сервер.

3.3 Интерфейса графической оболочке сервера

Графическая оболочка сервера максимально проста. В ней присутствует таблица с игроками и временем, за которое они справились с задачей. Так же присутствует кнопка, которая отчищает таблицу.

4 Описание программы

4.1 Интерфейс клиентской части

Интерфейс клиентской части системы программ представляет собой меню из трех кнопок и игровое поле, которое регулируется с помощью настроек.

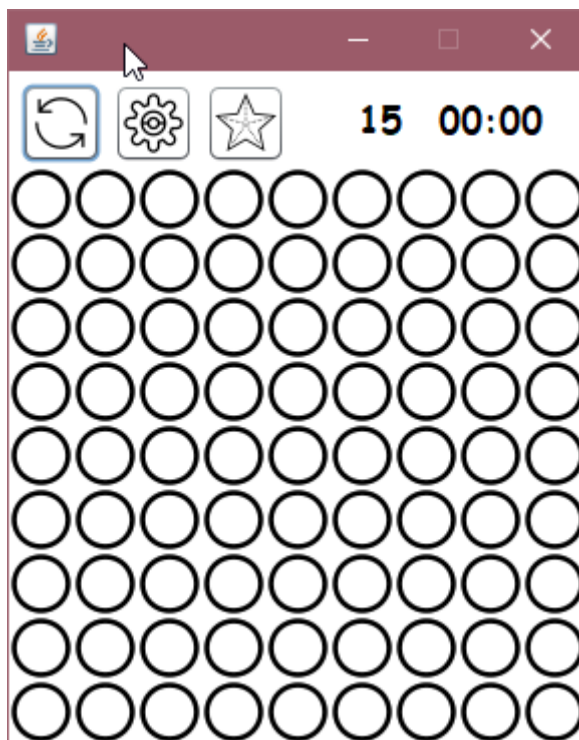


Рисунок 1 - графический интерфейс игры «Сапер»

Элементы интерфейса пронумерованы на рисунке 2.

1. Кнопка «Новая игра». Начинает игру оставляя то же количество бомб, но меняя их расположение и размер поля.
2. Кнопка «Настройки». Открывает меню настроек, в котором можно изменить размер поля и количество мин, а также выбрать готовый набор настроек для того, чтобы достижение игрока было занесено в список.
3. Кнопка «Рейтинг игроков». Список игроков, которые решили головоломку с готовым набором опций в меню «Настройки». Список сортируется по возрастанию.
4. Количество не найденных мин.

5. Таймер. Останавливается, когда игрок находит все мины. Используется как учетная единица в рейтинге игроков.
6. Игровое поле. Может регулироваться с помощью настроек. Главный элемент интерфейса, на котором протекает весь игровой процесс.

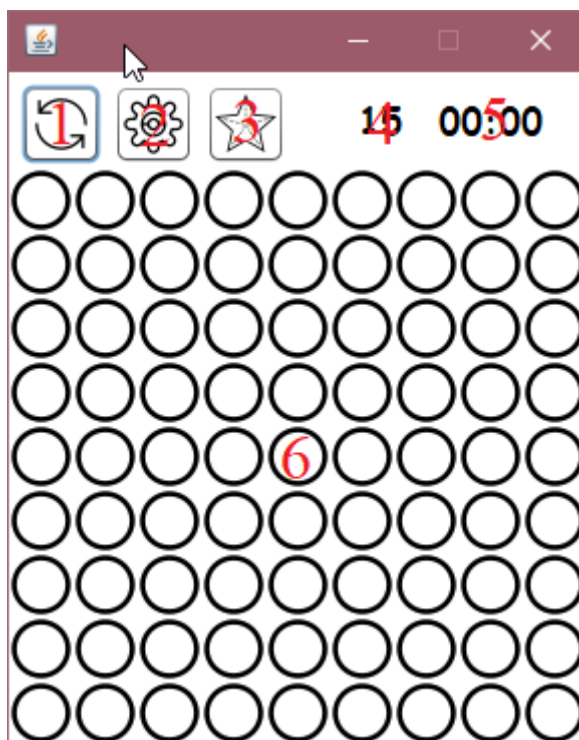


Рисунок 2 - пронумерованные элементы интерфейса

4.2 Интерфейс окна «Настройки»

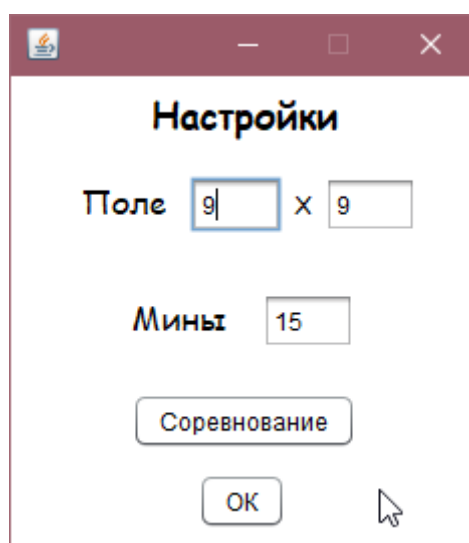


Рисунок 3 - меню настройки

В данном меню можно настроить высоту и ширину игрового поля, а также количество мин, которые будут размещены генератором случайных чисел на этом поле. Так же при нажатии на кнопку «Соревнование» устанавливаются заранее заготовленные настройки (поле 16 на 16, 40 мин). Если данные настройки будут установлены, и игрок решит головоломку, то его временной результат будет занесен в рейтинговую таблицу.

4.3 Интерфейс окна «Рейтинг игроков»

В окне «Рейтинг игроков» отображается список игроков, которые решили головоломку и за сколько им это удалось сделать при наборе настроек «Соревнование». Список игроков сортируется от самого короткого промежутка времени к самому длинному. Все данные отправляются на сервер. В противном случае если подключение к серверу отсутствует, то данные восстановить не удастся.

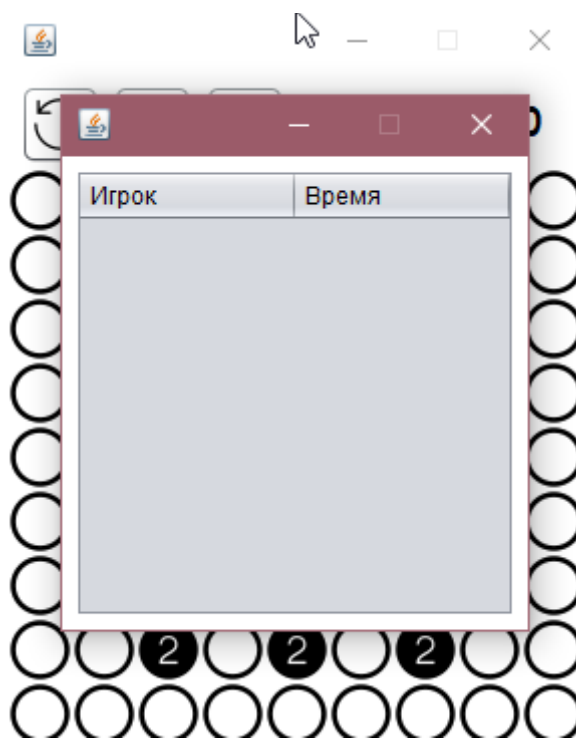


Рисунок 4 - интерфейс окна «Рейтинг игроков»

4.4 Интерфейс графической оболочки сервера

Интерфейс сервера практически полностью повторяет интерфейс окна «Рейтинг игроков» за исключением того, что в данном окне появилась кнопка очистить рейтинг, которая удаляет все данные о игроках.

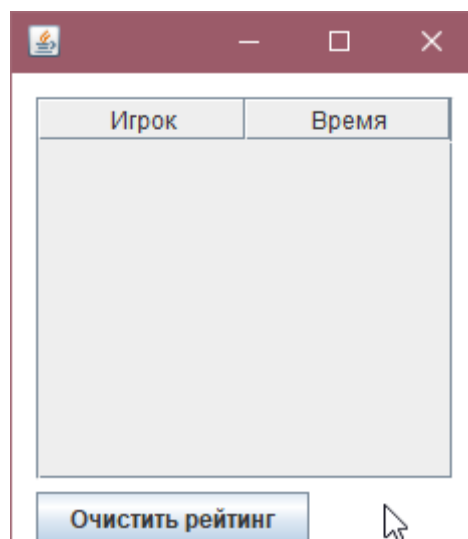


Рисунок 5- графическая оболочка сервера

ЗАКЛЮЧЕНИЕ

При выполнении данного курсового проекта были изучены принципы работы протокола TCP/IP. Были получены навыки разработки многомодульного приложения с пользовательским графическим интерфейсом и работы в среде NetBeans. Получены навыки разработки клиент-серверных приложений. А также улучшены навыки ООП и алгоритмизации. В курсовой работе применялись различные конструкции такие как интерфейсы, статические классы, рекурсивные алгоритмы. Было использовано многопоточное программирование, для разгрузки одного потока и более плавной работы графического интерфейса, а также параллельной обработки данных с нескольких клиентов на сервере.

В результате выполнения данной курсовой работы была разработана игра «Сапер» на языке программирования Java.

СПИСОК ЛИТЕРАТУРЫ

1. "Java. Методы программирования" И.Н Блинов, В.С. Романчик – Минск: УниверсалПресс, 2007 – 704 с.
2. "Java. Руководство для начинающих", Герберт Шилдт – ООО «Диалектика», 2019 – 816 с.
3. Форум программистов и сисадминов Киберфорум [Электронный ресурс]. Режим доступа: <http://www.cyberforum.ru/>, свободный (Дата обращения 29.04.2020).

Приложение А. Листинг программы клиента

Приложение А.1. Файл «Window.java»

```
package com.dmitr.minesweeper;

import com.dmitr.minesweeper.Interface.FrameInterface;
import com.dmitr.minesweeper.threads.ServerThread;

import java.awt.Color;
import java.awt.event.ActionEvent;
import javax.swing.JOptionPane;

/*
 * 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.
 */

/**
 *
 * @author dmitr
 */
public class Window extends javax.swing.JFrame implements
FrameInterface{

    private final FieldComponent mineField;

    private final int MIN_CELLS = 8;

    private final int WIDTH_DIALOG_WINDOW = 232;
    private final int HEIGHT_DIALOG_WINDOW = 237;

    private final int WIDTH_FRAME = 16;
    private final int HEIGHT_FRAME = 38;
    private final int HEIGHT_MENU = 48;

    private final int RATE_SIZE_CELLS = 16;
    private final int RATE_MINE_CELLS = 40;

    int widthCells = 9;
    int heightCells = 9;
    int minesCells = 15;

    ServerThread socketThread;

    public Window(){
        initComponents();
    }
}
```

```

        setSize(widthCells * FieldComponent.SIZE_CELL +
WIDTH_FRAME,
                heightCells * FieldComponent.SIZE_CELL +
HEIGHT_FRAME + HEIGHT_MENU);
        getContentPane().setBackground(Color.WHITE);
        setResizable(false);

        jFrame1.setSize(WIDTH_DIALOG_WINDOW + WIDTH_FRAME,
                HEIGHT_DIALOG_WINDOW + HEIGHT_FRAME);
        jFrame1.getContentPane().setBackground(Color.WHITE);

        jFrame2.setSize(WIDTH_DIALOG_WINDOW + WIDTH_FRAME,
                HEIGHT_DIALOG_WINDOW + HEIGHT_FRAME);
        jFrame2.getContentPane().setBackground(Color.WHITE);

        jFrame3.setSize(WIDTH_DIALOG_WINDOW + WIDTH_FRAME,
                HEIGHT_DIALOG_WINDOW + HEIGHT_FRAME);
        jFrame3.getContentPane().setBackground(Color.WHITE);

        mineField = new FieldComponent(widthCells, heightCells,
minesCells, this);
        mineField.setLocation(0, HEIGHT_MENU);
        add(mineField);

        socketThread = new ServerThread(jTable2, null);
        Thread thread = new Thread(socketThread);
        thread.start();

        setVisible(true);
    }

    /**
     * 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() {

        jFrame1 = new javax.swing.JFrame();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jTextField1 = new javax.swing.JTextField();
        jTextField2 = new javax.swing.JTextField();
        jTextField4 = new javax.swing.JTextField();
        jButton3 = new javax.swing.JButton();

```

```

jButton4 = new javax.swing.JButton();
jFrame2 = new javax.swing.JFrame();
jTextField3 = new javax.swing.JTextField();
jButton6 = new javax.swing.JButton();
jLabel7 = new javax.swing.JLabel();
jFrame3 = new javax.swing.JFrame();
jScrollPane2 = new javax.swing.JScrollPane();
jTable2 = new javax.swing.JTable();
jButton1 = new javax.swing.JButton();
jButton2 = new javax.swing.JButton();
jLabel11 = new javax.swing.JLabel();
jLabel6 = new javax.swing.JLabel();
jButton5 = new javax.swing.JButton();

jFrame1.setResizable(false);

jLabel2.setFont(new java.awt.Font("Comic Sans MS", 0,
16)); // NOI18N
jLabel2.setText("Поле");

jLabel3.setFont(new java.awt.Font("Comic Sans MS", 0,
12)); // NOI18N
jLabel3.setText("X");

jLabel4.setFont(new java.awt.Font("Comic Sans MS", 0,
16)); // NOI18N
jLabel4.setText("Мины");

jLabel5.setFont(new java.awt.Font("Comic Sans MS", 1,
18)); // NOI18N
jLabel5.setText("Настройки");

jTextField1.setBackground(new java.awt.Color(255, 255,
255));
jTextField1.setText("9");

jTextField2.setBackground(new java.awt.Color(255, 255,
255));
jTextField2.setText("9");

jTextField4.setBackground(new java.awt.Color(255, 255,
255));
jTextField4.setText("15");

jButton3.setBackground(new java.awt.Color(255, 255, 255));
jButton3.setText("OK");
jButton3.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent
evt) {
        jButton3ActionPerformed(evt);
    }
}

```

```

    });

    jButton4.setBackground(new java.awt.Color(255, 255, 255));
    jButton4.setText("Сопревождение");
    jButton4.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
evt) {
            jButton4ActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout jFrame1Layout = new
javax.swing.GroupLayout(jFrame1.getContentPane());
    jFrame1.getContentPane().setLayout(jFrame1Layout);
    jFrame1Layout.setHorizontalGroup(

jFrame1Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jFrame1Layout.createSequentialGroup()
            .addGap(0, 35, Short.MAX_VALUE)

.addGroup(jFrame1Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING)

.addGroup(jFrame1Layout.createSequentialGroup()
            .addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELA
TED)
                .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 46,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)
                .addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)
                .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE, 46,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(jFrame1Layout.createSequentialGroup()
            .addGap(25, 25, 25)

.addGroup(jFrame1Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING)
                .addComponent(jButton4)

```

```

.addGroup(jFrame1Layout.createSequentialGroup()
            .addComponent(jLabel4)
            .addGap(18, 18, 18)
            .addComponent(jTextField4,
javax.swing.GroupLayout.PREFERRED_SIZE, 46,
javax.swing.GroupLayout.PREFERRED_SIZE))))
        .addGap(30, 30, 30))
        .addGroup(jFrame1Layout.createSequentialGroup()

.addGroup(jFrame1Layout.createParallelGroup(javax.swing.GroupLayout
t.Alignment.LEADING)

.addGroup(jFrame1Layout.createSequentialGroup()
            .addGap(70, 70, 70)
            .addComponent(jLabel5))

.addGroup(jFrame1Layout.createSequentialGroup()
            .addGap(93, 93, 93)
            .addComponent(jButton3)))

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
    jFrame1Layout.setVerticalGroup(

jFrame1Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
        .addGroup(jFrame1Layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel5)
            .addGap(18, 18, 18)

.addGroup(jFrame1Layout.createParallelGroup(javax.swing.GroupLayout
t.Alignment.BASELINE)
            .addComponent(jLabel2)
            .addComponent(jLabel3)
            .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(30, 30, 30)

.addGroup(jFrame1Layout.createParallelGroup(javax.swing.GroupLayout
t.Alignment.BASELINE)
            .addComponent(jLabel4)
            .addComponent(jTextField4,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,

```

```

javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(22, 22, 22)
    .addComponent(jButton4)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELA
TED)
    .addComponent(jButton3)

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );

jFrame2.setResizable(false);

jTextField3.setBackground(new java.awt.Color(255, 255,
255));
jTextField3.setFont(new java.awt.Font("Comic Sans MS", 0,
16)); // NOI18N

jTextField3.setHorizontalAlignment(javax.swing.JTextField.CENTER);
jTextField3.setText("Введите имя");

jButton6.setBackground(new java.awt.Color(255, 255, 255));
jButton6.setText("OK");
jButton6.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent
evt) {
        jButton6ActionPerformed(evt);
    }
});

jLabel7.setBackground(new java.awt.Color(255, 255, 255));
jLabel7.setFont(new java.awt.Font("Comic Sans MS", 1,
18)); // NOI18N

jLabel7.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel7.setText("Ваш результат 00:00");

javax.swing.GroupLayout jFrame2Layout = new
javax.swing.GroupLayout(jFrame2.getContentPane());
jFrame2.getContentPane().setLayout(jFrame2Layout);
jFrame2Layout.setHorizontalGroup(

jFrame2Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
    .addGroup(jFrame2Layout.createSequentialGroup())
        .addGap(37, 37, 37)
        .addComponent(jTextField3,
javax.swing.GroupLayout.PREFERRED_SIZE, 159,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(jFrame2Layout.createSequentialGroup())

        .addGroup(jFrame2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(jFrame2Layout.createSequentialGroup()
            .addGap(25, 25, 25)
            .addComponent(jLabel17))

        .addGroup(jFrame2Layout.createSequentialGroup()
            .addGap(89, 89, 89)
            .addComponent(jButton6))
            .addContainerGap(16, Short.MAX_VALUE))
    );
    jFrame2Layout.setVerticalGroup(

jFrame2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jFrame2Layout.createSequentialGroup()
            .addGap(80, 80, 80)
            .addComponent(jLabel17)
            .addGap(18, 18, 18)
            .addComponent(jTextField3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 44, Short.MAX_VALUE)
            .addComponent(jButton6)
            .addContainerGap())
    );

    jFrame3.setResizable(false);

    jTable2.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {

            },
            new String [] {
                "Игрок", "Время"
            }
        ) {
        Class[] types = new Class [] {
            java.lang.String.class, java.lang.String.class
        };
        boolean[] canEdit = new boolean [] {
            false, false
        };

        public Class getColumnClass(int columnIndex) {

```

```

        return types [columnIndex];
    }

    public boolean isCellEditable(int rowIndex, int
columnIndex) {
        return canEdit [columnIndex];
    }
});
jScrollPane2.setViewportViewView(jTable2);
if (jTable2.getColumnModel().getColumnCount() > 0) {

jTable2.getColumnModel().getColumn(0).setResizable(false);

jTable2.getColumnModel().getColumn(1).setResizable(false);
}

    javax.swing.GroupLayout jFrame3Layout = new
javax.swing.GroupLayout(jFrame3.getContentPane());
    jFrame3.getContentPane().setLayout(jFrame3Layout);
    jFrame3Layout.setHorizontalGroup(

jFrame3Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
        .addGroup(jFrame3Layout.createSequentialGroup()
            .addGap()
            .addComponent(jScrollPane2,
javax.swing.GroupLayout.DEFAULT_SIZE, 388, Short.MAX_VALUE)
            .addGap())
        );
    jFrame3Layout.setVerticalGroup(

jFrame3Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
        .addGroup(jFrame3Layout.createSequentialGroup()
            .addGap()
            .addComponent(jScrollPane2,
javax.swing.GroupLayout.DEFAULT_SIZE, 225, Short.MAX_VALUE)
            .addGap())
        );

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE
);
setBackground(new java.awt.Color(0, 0, 0));

jButton1.setBackground(new java.awt.Color(255, 255, 255));
jButton1.setFont(new java.awt.Font("Comic Sans MS", 1,
14)); // NOI18N
jButton1.setIcon(new
javax.swing.ImageIcon("src/new_game.png")); // NOI18N
jButton1.addActionListener(new
java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent
evt) {
            JButton1ActionPerformed(evt);
        }
    });

    JButton2.setBackground(new java.awt.Color(255, 255, 255));
    JButton2.setFont(new java.awt.Font("Comic Sans MS", 1,
12)); // NOI18N
    JButton2.setIcon(new
javax.swing.ImageIcon("src/setting.png")); // NOI18N
    JButton2.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
evt) {
            JButton2ActionPerformed(evt);
        }
    });

    JLabel1.setFont(new java.awt.Font("Comic Sans MS", 1,
18)); // NOI18N

    JLabel1.setHorizontalAlignment(javax.swing.SwingConstants.TRAILING
);
    JLabel1.setText("00:00");

    JLabel6.setFont(new java.awt.Font("Comic Sans MS", 1,
18)); // NOI18N

    JLabel6.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
    JLabel6.setText("0");

    JButton5.setBackground(new java.awt.Color(255, 255, 255));
    JButton5.setIcon(new
javax.swing.ImageIcon("src/top.png")); // NOI18N
    JButton5.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
evt) {
            JButton5ActionPerformed(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()
            .addContainerGap()

```

```

        .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)
        .addComponent(jButton2,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)
        .addComponent(jButton5,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D, 85, Short.MAX_VALUE)
        .addComponent(jLabel6)
        .addGap(18, 18, 18)
        .addComponent(jLabel1)
        .addGap(22, 22, 22))
);
layout.setVerticalGroup(

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

        .addGroup(layout.createSequentialGroup()
                .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                .addComponent(jButton5,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jButton2,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGroup(layout.createSequentialGroup()
                        .addGap(5, 5, 5)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)
                        .addComponent(jLabel1)
                        .addComponent(jLabel6)))
                .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(93, Short.MAX_VALUE))
);

pack();

```

```

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

private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    mineField.newGame();
}

private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    jFrame1.setVisible(true);
}

private void jButton3ActionPerformed(ActionEvent evt) {
    if (applySetting())
    {
        jFrame1.setVisible(false);
        mineField.newGame(widthCells, heightCells,
minesCells);

        setSize(widthCells * FieldComponent.SIZE_CELL +
WIDTH_FRAME,
                heightCells * FieldComponent.SIZE_CELL +
HEIGHT_FRAME + HEIGHT_MENU);
    }
}

private void jButton4ActionPerformed(ActionEvent evt) {
    jFrame1.setVisible(false);
    mineField.newGame(RATE_SIZE_CELLS, RATE_SIZE_CELLS,
RATE_MINE_CELLS);

    setSize(RATE_SIZE_CELLS * FieldComponent.SIZE_CELL +
WIDTH_FRAME,
            RATE_SIZE_CELLS * FieldComponent.SIZE_CELL +
HEIGHT_FRAME + HEIGHT_MENU);
}

private void
jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    jFrame3.setVisible(true);
}

private void
jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

    jFrame2.setVisible(false);
    Result result = new Result();
    String str = jTextField3.getText();
    if (str.equals("Введите имя") || str.equals(""))
        result.name = "no name";
    else
        result.name = str;
}

```

```

        result.time = mineField.getTime();
        //socketThread.outputData(result);

        ServerThread serverThread = new ServerThread(jTable2,
result);
        Thread thread = new Thread(serverThread);
        thread.start();
        jFrame3.setVisible(true);
    }

    /**
     * @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(Window.class.getName()).log(jav
a.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Window.class.getName()).log(jav
a.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Window.class.getName()).log(jav
a.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Window.class.getName()).log(jav
a.util.logging.Level.SEVERE, null, ex);
        }
    }
    //</editor-fold>

    /* Create and display the form */

```

```

        java.awt.EventQueue.invokeLater(new Runnable() {
            @Override
            public void run() {
                new Window();
            }
        });
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JButton jButton4;
    private javax.swing.JButton jButton5;
    private javax.swing.JButton jButton6;
    private javax.swing.JFrame jFrame1;
    private javax.swing.JFrame jFrame2;
    private javax.swing.JFrame jFrame3;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JTable jTable2;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    private javax.swing.JTextField jTextField3;
    private javax.swing.JTextField jTextField4;

    public boolean applySetting() {
        try
        {
            int newWidthCell =
Integer.parseInt(jTextField1.getText());
            int newHeightCell =
Integer.parseInt(jTextField2.getText());
            int newMineCells =
Integer.parseInt(jTextField4.getText());

            if(newWidthCell > MIN_CELLS && newHeightCell >
MIN_CELLS && newMineCells < newWidthCell * newHeightCell){
                widthCells = newWidthCell;
                heightCells = newHeightCell;
                minesCells = newMineCells;

                return true;
            }
            else
                {

```

```

        JOptionPane.showMessageDialog(null,
            "Минимальное поле 9x9,\nколичество мин не
должно превышать размер поля");
        return false;
    }
}
catch(Exception e) {
    JOptionPane.showMessageDialog(null, "В полях должны
быть\nтолько целые числа");
    return false;
}
}

@Override
public void outputMessage(String message) {
    JOptionPane.showMessageDialog(null, message);
}

@Override
public void openRating(String time) {
    jLabel7.setText("Ваш результат " + time);
    jFrame2.setVisible(true);
}

@Override
public void setTimer(String time) {
    jLabel11.setText(time);
}

@Override
public void setLeftFlags(String flags) {
    jLabel6.setText(flags);
}

@Override
public void closeWindows() {
    jFrame1.setVisible(false);
    jFrame2.setVisible(false);
    jFrame3.setVisible(false);
}
}

```

Приложение А.2. Файл «Cell.java»

```
package com.dmitr.minesweeper; /*
 * 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.
 */

/**
 *
 * @author dmitr
 */

public class Cell {
    public boolean mine;
    public boolean flag;
    public boolean open;

    public Cell()
    {
        this.mine = false;
        this.flag = false;
        this.open = false;
    }

    boolean setmine()
    {
        if(mine)
            return false;
        else
        {
            mine = true;
            return true;
        }
    }
}
```

Приложение А.3. Файл «Field.java»

```
package com.dmitr.minesweeper; /*
 * 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.
 */

/**
 *
 * @author dmitr
 */
```

```

    */
    public class Field {
        Cell arrayCells[][];

        int width;
        int height;

        public Field(int width, int height, int mine, int banX, int
        banY){
            this.width = width;
            this.height = height;

            arrayCells = new Cell[width][height];
            for(int i = 0; i < width; i++)
            {
                for(int j = 0; j < height; j++)
                {
                    arrayCells[i][j] = new Cell();
                }
            }

            int mineX;
            int mineY;

            for(int i = 0; i < mine; i++)
            {
                boolean mk;
                do{
                    mineX = (int) (Math.random()*width);
                    mineY = (int) (Math.random()*height);
                    if(banX != mineX || banY != mineY)
                        mk = arrayCells[mineX][mineY].setmine();
                    else
                        mk = false;
                }while(!mk);
            }
        }

        public String leftClick(int x, int y) throws
        CanNotOpenException{
            if (arrayCells[x][y].mine)
            {
                arrayCells[x][y].open =true;
                return "mine";
            }
            else if (arrayCells[x][y].flag || arrayCells[x][y].open)
                throw new CanNotOpenException();
            else
            {
                arrayCells[x][y].open =true;
                return calcMines(x, y);
            }
        }
    }

```

```

    }

    public boolean rightClick(int x, int y) throws
CanNotOpenException{
        if (arrayCells[x][y].open)
            throw new CanNotOpenException();

        arrayCells[x][y].flag = !arrayCells[x][y].flag;

        return arrayCells[x][y].mine;
    }

    public void outputMines(String imagesArray[][]){
        for(int i = 0; i < width; i++)
            for(int j = 0; j < height; j++)
                if(arrayCells[i][j].mine)
                    imagesArray[i][j] = "mine";
    }

    private String calcMines(int x, int y){
        int k = 0;
        for(int i=-1; i < 2; i++)
            for(int j=-1; j < 2; j++)
                try{
                    if(arrayCells[x+i][y+j].mine)
                        k++;
                }catch(Exception e){}
        return String.valueOf(k);
    }

    public String calcMinesWithoutFlags(int x, int y) throws
CanNotOpenException{
        if (arrayCells[x][y].flag || arrayCells[x][y].open)
            throw new CanNotOpenException();
        arrayCells[x][y].open = true;

        return calcMines(x, y);
    }
}

```

Приложение A.4. Файл «FieldComponent.java»

```

package com.dmitr.minesweeper;

import com.dmitr.minesweeper.Interface.FrameInterface;
import java.awt.Graphics;
import java.awt.Image;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.io.File;
import java.io.IOException;

```

```

import java.util.HashMap;
import java.util.Map;
import java.util.Timer;
import java.util.TimerTask;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.imageio.ImageIO;
import javax.swing.JPanel;

public class FieldComponent extends JPanel{

    public static final int SIZE_CELL = 32;

    private final FrameInterface mainWindow;
    private Field field;

    private Timer timer;
    private int seconds;

    private String[][] imagesArray;
    private final Map<String, Image> imagesButtons;

    private int mineCells;
    private int widthCells;
    private int heightCells;

    private int trueFlags;
    private int leftFlags;

    private boolean endGame;
    private boolean press;
    private boolean first;

    private int pressX;
    private int pressY;

    public FieldComponent(int widthCells, int heightCells, int
mineCells, FrameInterface mainWindow) {

        this.mainWindow = mainWindow;
        timer = new Timer();

        imagesButtons = new HashMap<>();

        try {
            imagesButtons.put("simple",
                ImageIO.read(new File("src/simple.png")));
            imagesButtons.put("mine",
                ImageIO.read(new File("src/mine.png")));
            imagesButtons.put("flag",

```

```

        ImageIO.read(new File("src/flag.png")));
imagesButtons.put("0",
        ImageIO.read(new File("src/0.png")));
imagesButtons.put("1",
        ImageIO.read(new File("src/1.png")));
imagesButtons.put("2",
        ImageIO.read(new File("src/2.png")));
imagesButtons.put("3",
        ImageIO.read(new File("src/3.png")));
imagesButtons.put("4",
        ImageIO.read(new File("src/4.png")));
imagesButtons.put("5",
        ImageIO.read(new File("src/5.png")));
imagesButtons.put("6",
        ImageIO.read(new File("src/6.png")));
imagesButtons.put("7",
        ImageIO.read(new File("src/7.png")));
imagesButtons.put("8",
        ImageIO.read(new File("src/8.png")));
} catch (IOException ex) {
    Logger.getLogger(Window.class.getName()).
        log(Level.SEVERE, null, ex);
}

addMouseListener(new MouseAdapter() {

    @Override
    public void mousePressed(MouseEvent e) {
        pressX = e.getX() / SIZE_CELL;
        pressY = e.getY() / SIZE_CELL;
        press = true;
    }

    @Override
    public void mouseReleased(MouseEvent e) {
        if(press && pressX == e.getX() / SIZE_CELL &&
pressY == e.getY() / SIZE_CELL && !endGame) {

            if(first)
                firstClick();

            if (e.getButton() == MouseEvent.BUTTON1)
                leftPress();

            if (e.getButton() == MouseEvent.BUTTON3)
                rightPress();
        }

        repaint();
        press = false;
    }
}

```

```

    });

    newGame(widthCells, heightCells, mineCells);
}

    public final void newGame(int widthCells, int heightCells, int
mineCells) {
        this.widthCells = widthCells;
        this.heightCells = heightCells;
        this.mineCells = mineCells;
        newGame();
    }

    public void newGame() {
        mainWindow.closeWindows();
        leftFlags = mineCells;
        endGame = false;
        seconds = 0;

        try {
            timer.cancel();
        }
        catch(Exception e)
        {
            Logger.getLogger(Window.class.getName()).
                log(Level.SEVERE, null, e);
        }

        mainWindow.setTimer("00:00");
        mainWindow.setLeftFlags(String.format("%d", mineCells));

        trueFlags = 0;

        imagesArray = new String[widthCells][heightCells];

        for(int i = 0; i < widthCells; i++)
            for(int j = 0; j < heightCells; j++)
                imagesArray[i][j] = "simple";

        super.setSize(SIZE_CELL * widthCells, SIZE_CELL *
heightCells);
        first = true;

        repaint();
    }

    private void openGlade(int x, int y) throws
CanNotOpenException {
        for(int i=-1; i < 2; i++)
            for(int j=-1; j < 2; j++)
                try {
                    imagesArray[x + i][y + j] =

```

```

field.calcMinesWithoutFlags(x + i, y + j);

        if ("0".equals(imagesArray[x + i][y + j]))
            openGlade(x + i, y + j);

    } catch (Exception e) {
        Logger.getLogger(Window.class.getName()).
            log(Level.SEVERE, null, e);
    }
}

private void firstClick() {
    field = new Field(widthCells, heightCells, mineCells,
pressX, pressY);
    first = false;

    TimerTask repeatedFunc = new TimerTask() {
        @Override
        public void run() {
            mainWindow.setTimer(String.format("%02d:%02d",
seconds / 60, seconds % 60));
            seconds++;
        }
    };

    timer = new Timer();
    timer.scheduleAtFixedRate(repeatedFunc, 0, 1000);
}

private void lastClick(String msg) {
    repaint();
    endGame = true;

    try {
        timer.cancel();
    }
    catch (Exception e) {
        Logger.getLogger(Window.class.getName()).
            log(Level.SEVERE, null, e);
    }

    mainWindow.outputMessage(msg);
}

private void rightPress() {
    try {
        if (field.rightClick(pressX, pressY)) {
            if ("simple".equals(imagesArray[pressX][pressY]))
{
                if (leftFlags > 0) {
                    trueFlags++;
                    leftFlags--;

```

```

        imagesArray[pressX][pressY] = "flag";

mainWindow.setLeftFlags(String.format("%d", leftFlags));

        if (trueFlags == mineCells) {
            lastClick("Вы выиграли!");

            if (widthCells == 9 && heightCells == 9
&&
                mineCells == 1)

mainWindow.openRating(String.format("%02d:%02d",
seconds / 60, seconds %
60));
        }
    }
    } else {
        trueFlags--;
        leftFlags++;

        imagesArray[pressX][pressY] = "simple";
        mainWindow.setLeftFlags(String.format("%d",
leftFlags));
    }
    } else {
        if ("simple".equals(imagesArray[pressX][pressY]))
{
            if (leftFlags > 0) {
                leftFlags--;

                imagesArray[pressX][pressY] = "flag";

mainWindow.setLeftFlags(String.format("%d", leftFlags));
            }
        } else {
            leftFlags++;

            imagesArray[pressX][pressY] = "simple";
            mainWindow.setLeftFlags(String.format("%d",
leftFlags));
        }
    }
} catch (CanNotOpenException ex) {
    Logger.getLogger(Window.class.getName()).
        log(Level.SEVERE, null, ex);
}

}

private void leftPress() {
    try {

```

```

        imagesArray[pressX][pressY] = field.leftClick(pressX,
pressY);
        if ("mine".equals(imagesArray[pressX][pressY])) {
            field.outputMines(imagesArray);
            lastClick("Бум! Вы проиграли!");
        }

        if ("0".equals(imagesArray[pressX][pressY])) {
            openGlade(pressX, pressY);
        }
    } catch (CanNotOpenException ex) {
        Logger.getLogger(Window.class.getName()).
            log(Level.SEVERE, null, ex);
    }
}

@Override
public void paintComponent(Graphics g) {
    g.clearRect(0, 0, widthCells * SIZE_CELL, heightCells *
SIZE_CELL);

    for(int i = 0; i < widthCells; i++)
        for(int j = 0; j < heightCells; j++)
            g.drawImage(imagesButtons.get(imagesArray[i][j]),
i * SIZE_CELL, j * SIZE_CELL, null);
}

public int getTime(){
    return seconds;
}
}

```

Приложение A.5. Файл «Result.java»

```

package com.dmitr.minesweeper;

import java.io.Serializable;

public class Result implements Serializable {
    public String name = "no name";
    public int time = 0;

    public Result(String name, int time){
        this.name = name;
        this.time = time;
    }

    public Result() { }
}

```

Приложение А.6. Файл «ListResult.java»

```
package com.dmitr.minesweeper;

import javax.swing.*;
import java.io.Serializable;
import java.util.LinkedList;

public class ListResult implements Serializable {

    public static LinkedList<Result> ratingList = new
LinkedList<Result>();

    ListResult()
    {
    }

    public static void add(Result result) {
        int i = 0;
        for(; (i < ratingList.size()) && comparator(result,
ratingList.get(i)); i++);
        ratingList.add(i, result);
    }

    private static boolean comparator(Result result, Result get) {
        return result.time < get.time;
    }

    public Result get(int i) {
        return ratingList.get(i);
    }

    public static void updateTable(JTable table)
    {
        try{
            javax.swing.table.DefaultTableModel model =
                (javax.swing.table.DefaultTableModel)
table.getModel();
            model.setRowCount(0);

            for (int i = 0; i < ratingList.size(); i++)
                model.addRow(new Object[]{ratingList.get(i).name,
                    String.format("%02d:%02d",
ratingList.get(i).time / 60,
ratingList.get(i).time % 60)});

        } catch(Exception e) { }
    }
}
```

Приложение A.7. Файл «CanNotOpenException.java»

```
package com.dmitr.minesweeper;

public class CanNotOpenException extends Exception {
    public CanNotOpenException ()
    {

    };
}
```

Приложение A.8. Файл «ServerThread.java»

```
package com.dmitr.minesweeper.threads;

import com.dmitr.minesweeper.Result;
import com.dmitr.minesweeper.ListResult;
import com.dmitr.minesweeper.Window;

import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.Socket;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.*;

public class ServerThread implements Runnable {

    private final Object lockIO = new Object();
    private Socket socket;
    private Result result;
    private JTable table;

    public ServerThread(JTable table, Result result) {
        this.table = table;
        this.result = result;
    }

    @Override
    public void run() {

        try {
            socket = new Socket("127.0.0.1", 5565);

            ObjectOutputStream outputStream =
                new
```

```

OutputStream(socket.getOutputStream());

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

    boolean resultOut = result != null;

    if (resultOut)
    {
        outputStream.writeBoolean(true);
        outputStream.flush();
        outputStream.writeUTF(result.name);
        outputStream.flush();
        outputStream.writeInt(result.time);
        outputStream.flush();

    } else {
        outputStream.writeBoolean(false);
        outputStream.flush();

    }

    resultOut = inputStream.readBoolean();
    String name;
    int time;
    ListResult.ratingList.clear();

    while(resultOut){
        name = inputStream.readUTF();
        time = inputStream.readInt();

        ListResult.add(new Result(name, time));

        resultOut = inputStream.readBoolean();
    }

    ListResult.updateTable(table);

    outputStream.close();
    inputStream.close();
    socket.close();

} catch (IOException ex) {
    Logger.getLogger(Window.class.getName()).
        log(Level.SEVERE, null, ex);
}
}
}

```

Приложение Б. Листинг программы сервера

Приложение Б.1. Файл «Window.java»

```
package com.dmitr.kr_server;

import com.dmitr.kr_server.threads.ServerThread;
import java.awt.Color;

public class Window extends javax.swing.JFrame {

    ServerThread serverThread;

    public Window() {
        initComponents();

        setSize(232 + 16, 237 + 38);
        getContentPane().setBackground(Color.WHITE);

        serverThread = new ServerThread(jTable2);
        Thread thread = new Thread(serverThread);
        thread.start();
    }

    /**
     * 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() {

        jScrollPane2 = new javax.swing.JScrollPane();
        jTable2 = new javax.swing.JTable();
        jButton1 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE
        );

        jTable2.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {

            },
            new String [] {
                "Игрок", "Время"
            }
        ));
    }
}
```

```

    }
    ) {
        Class[] types = new Class [] {
            java.lang.String.class, java.lang.String.class
        };
        boolean[] canEdit = new boolean [] {
            false, false
        };

        public Class getColumnClass(int columnIndex) {
            return types [columnIndex];
        }

        public boolean isCellEditable(int rowIndex, int
columnIndex) {
            return canEdit [columnIndex];
        }
    });
    jScrollPane2.setViewportViewView(jTable2);

    jButton1.setText("Очистить рейтинг");
    jButton1.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
evt) {
            jButton1ActionPerformed(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(15, 15, 15)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jScrollPane2,
javax.swing.GroupLayout.DEFAULT_SIZE, 318, Short.MAX_VALUE)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jButton1)
                    .addGap(15, 15, 15)
                    .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE, 150, Short.MAX_VALUE))
                .addGap(15, 15, 15))
            .addContainerGap(15, true))
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(15, 15, 15)
            .addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT_SIZE, 318, Short.MAX_VALUE)
            .addContainerGap(15, true))
    );
    layout.setHorizontalGroup(

```

```

        .addGroup(layout.createSequentialGroup())
            .addContainerGap()
            .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE, 191,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jButton1)
            .addContainerGap()
    );

    pack();
} // </editor-fold> // GEN-END: initComponents

private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_jButton1ActionPerformed
    ListResult.ratingList.clear();

    ListResult.updateTable(jTable2);
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Window().setVisible(true);
        }
    });
}

// Variables declaration - do not modify // GEN-BEGIN:variables
private javax.swing.JButton jButton1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JTable jTable2;
// End of variables declaration // GEN-END:variables
}

```

Приложение Б.2. Файл «Result.java»

```

package com.dmitr.kr_server;

import java.io.Serializable;

public class Result implements Serializable {
    public String name = "no name";
    public int time = 0;

    public Result(String name, int time){

```

```

        this.name = name;
        this.time = time;
    }
}

```

Приложение Б.3. Файл «ListResult.java»

```

package com.dmitr.kr_server;

import javax.swing.*;
import java.io.Serializable;
import java.util.LinkedList;

public class ListResult implements Serializable {

    public static LinkedList<Result> ratingList = new
LinkedList<>();

    public static void add(Result result) {
        int i = 0;
        for(; (i > ratingList.size()) && comparator(result,
ratingList.get(i)); i++);
            ratingList.add(i, result);
    }

    private static boolean comparator(Result result, Result get) {
        return result.time < get.time;
    }

    public static void updateTable(JTable table)
    {
        try{
            javax.swing.table.DefaultTableModel model =
                (javax.swing.table.DefaultTableModel)
table.getModel();
            model.setRowCount(0);

            for (int i = 0; i < ratingList.size(); i++)
                model.addRow(new Object[]{ratingList.get(i).name,
                    String.format("%02d:%02d",
ratingList.get(i).time / 60,
ratingList.get(i).time % 60)});

        } catch (Exception e) { }
    }

    public Result get(int i) {
        return ratingList.get(i);
    }

    public int size() {

```

```

        return ratingList.size();
    }

    public void clear() {
        ratingList.clear();
    }
}

```

Приложение Б.4. Файл «ServerThread.java»

```

package com.dmitr.kr_server.threads;

import java.io.IOException;
import java.net.ServerSocket;
import javax.swing.JTable;

public class ServerThread implements Runnable{
    private ServerSocket serverSocket;
    public JTable table;

    public ServerThread(JTable table)
    {
        this.table = table;
    }

    @Override
    public void run() {
        try {
            serverSocket = new ServerSocket(5565);
            while(true) {
                ClientHandlerThread clientHandlerThread = new
ClientHandlerThread(serverSocket.accept(), table);
                Thread thread = new Thread(clientHandlerThread);
                thread.start();
            }
        } catch (IOException ex) {}
    }
}

```

Приложение Б.5. Файл «ClientHandlerThead.java»

```
package com.dmitr.kr_server.threads;

import com.dmitr.kr_server.ListResult;
import com.dmitr.kr_server.Result;
import com.dmitr.kr_server.Window;

import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.Socket;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JTable;

public class ClientHandlerThread implements Runnable{

    static private Object lockIO = new Object();

    private JTable table;
    private Result result;

    Socket socket;

    ClientHandlerThread(Socket socket, JTable table) {
        this.socket = socket;
        this.table = table;
    }

    @Override
    public void run() {
        try {

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

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

            boolean ready = inputStream.readBoolean();

            if (ready) {
                String name = inputStream.readUTF();
                int time = inputStream.readInt();

                Result result = new Result(name, time);
                ListResult.add(result);
            }
        }
    }
}
```

```

        ListResult.updateTable(table);
    }

    for (int i = 0; i < ListResult.ratingList.size();
i++) {
        outputStream.writeBoolean(true);
        outputStream.flush();

        outputStream.writeUTF(ListResult.ratingList.get(i).name);
        outputStream.flush();

        outputStream.writeInt(ListResult.ratingList.get(i).time);
        outputStream.flush();
    }

    outputStream.writeBoolean(false);
    outputStream.flush();

    outputStream.close();
    inputStream.close();
    socket.close();

} catch (Exception ex) {
    Logger.getLogger(Window.class.getName()).
        log(Level.SEVERE, null, ex);
}

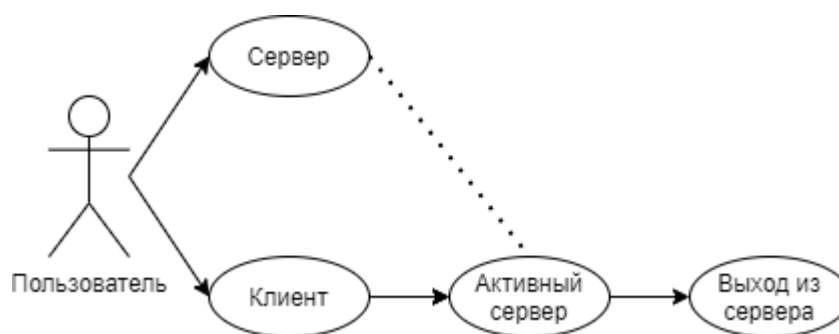
}

```

Приложение В. UML-диаграммы

Приложение В.1. UML-диаграмма вариантов использования

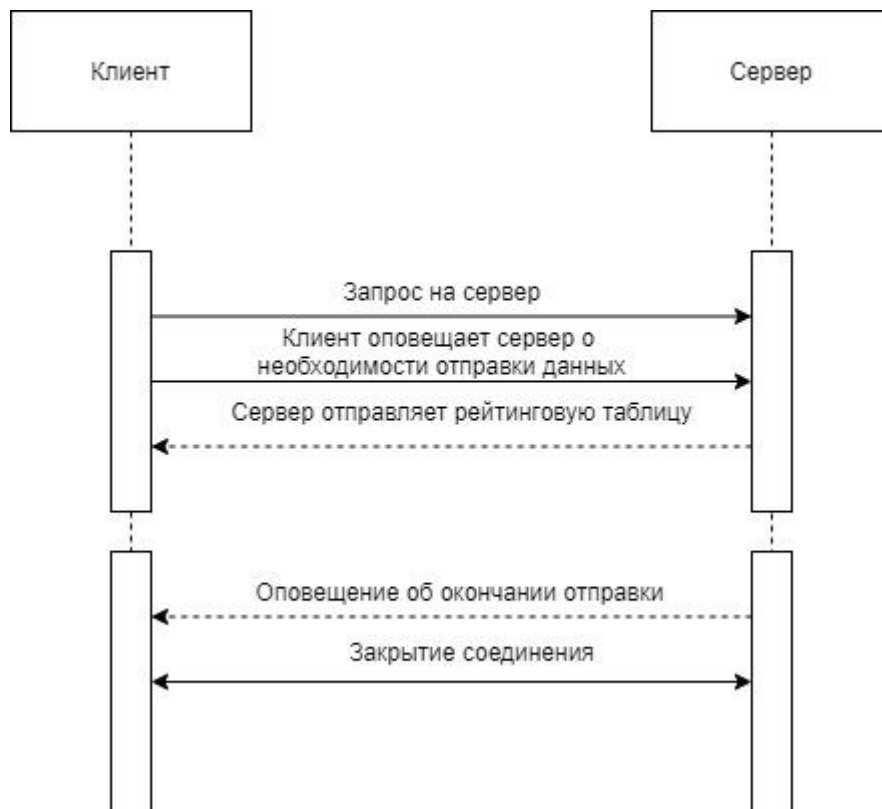
UML-диаграмма вариантов использования представлена на рисунке 9.



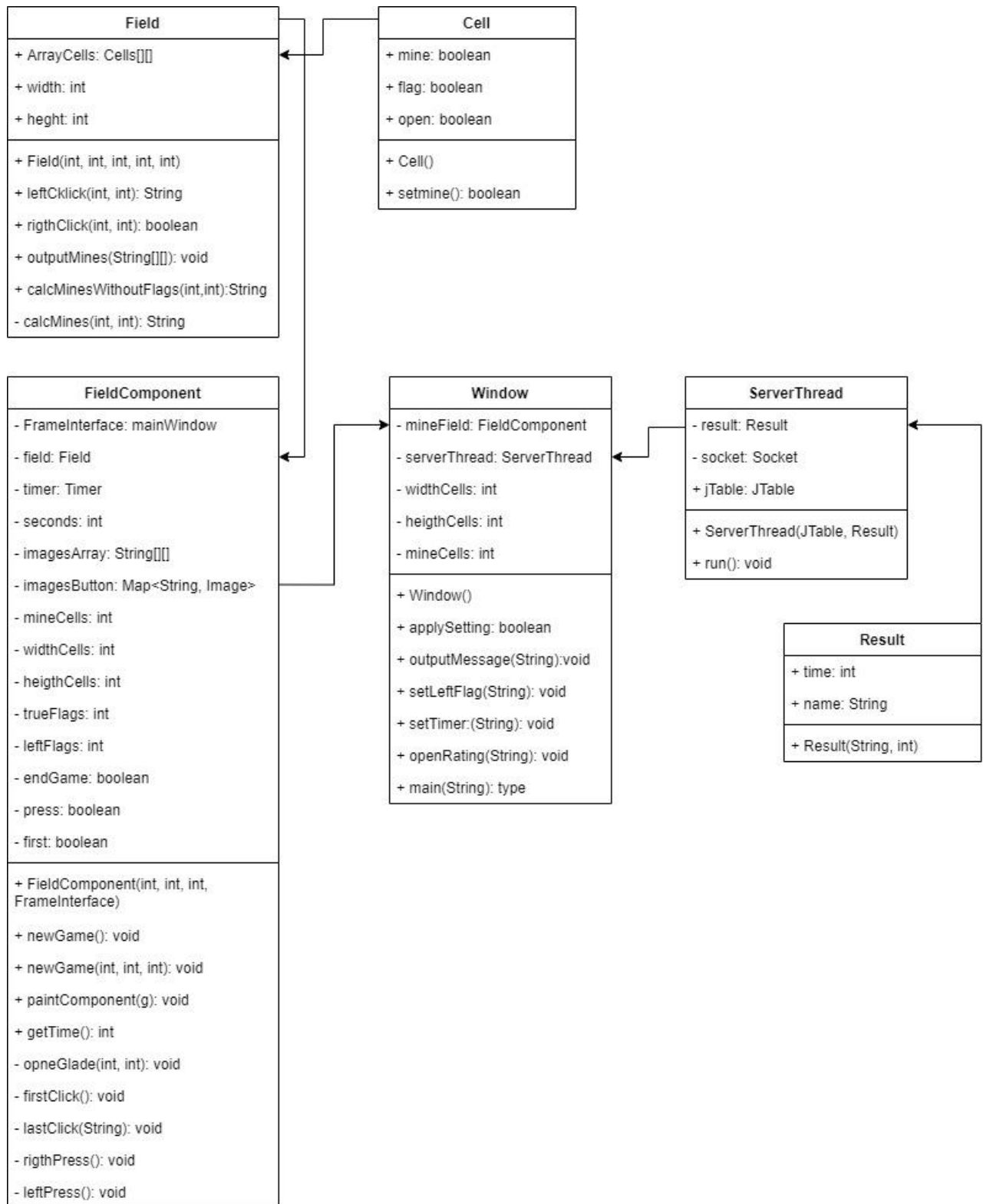
Приложение В.2.

UML- диаграмма последовательности

UML- диаграмма последовательности представлена на рисунке 10.



Приложение В.3. UML-диаграмма классов



Приложение В.4. UML-диаграмма развертывания

