Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**Иркутский национальный исследовательский технический университет**

|  |
| --- |
| Институт информационных технологий и анализа данных |
| наименование института |
| Кафедра вычислительной техники |
| наименование кафедры |

|  |
| --- |
| **Отчет** |
| по лабораторной работе №5 по дисциплине «Разработка приложения с оконным графическим пользовательским интерфейсом» |
| наименование темы  Вариант №29 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент |  | АСУб-19-1 |  | C:\Users\Kinzel\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1.png |  | Янхаев Д.В. |
|  |  | шифр |  | подпись |  | И.О. Фамилия |
| Проверил |  |  |  |  |  | Маланова Т.В. |
|  |  |  |  | подпись |  | И.О. Фамилия |
| Работа защищена с оценкой | | | |  | | |

Иркутск 2020 г.

**Содержание**

[1 Постановка задачи 3](#_Toc59591548)

[2 Проектирование графического интерфейса пользователя 4](#_Toc59591549)

[3 Описание структуры классов 6](#_Toc59591550)

[4 Таблица тестов 7](#_Toc59591551)

[5 Результаты тестирования 8](#_Toc59591552)

[6 Листинг исходного кода 12](#_Toc59591553)

[Список использованных источников 17](#_Toc59591554)

1 Постановка задачи

Для выполнения задания необходимо разработать графический пользовательский интерфейс средствами java для работы с классами реализованными при выполнении либо 3-ей, либо 4-ой лабораторной работы.

Графический пользовательский интерфейс должен предоставлять возможности работы со всеми возможными функциями программы (ввод и вывод данных в соответствующей форме, изменение условий выполнения задания, работа с файлами и т.д.). Состав используемых компонентов определяется самостоятельно, но для них должны быть предусмотрены соответствующие планировщики раскладки компонентов в основных контейнерах.

Классы, реализующие графический пользовательский интерфейс, должны содержать только обращения к методам классов для выполнения заданий. Для этого может понадобиться модифицировать программный код из предыдущих лабораторных работ.

2 Проектирование графического интерфейса пользователя

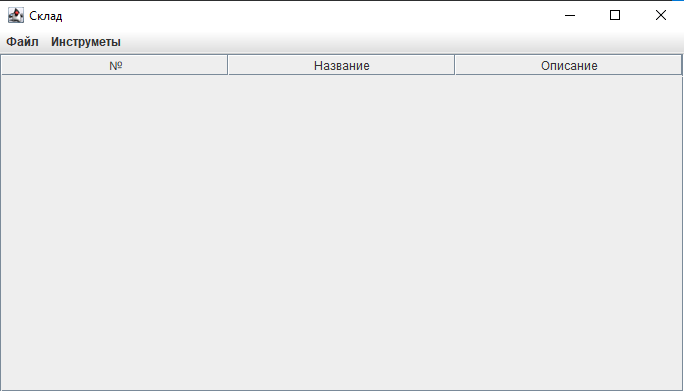


Рис. 1 – Главное окно приложения

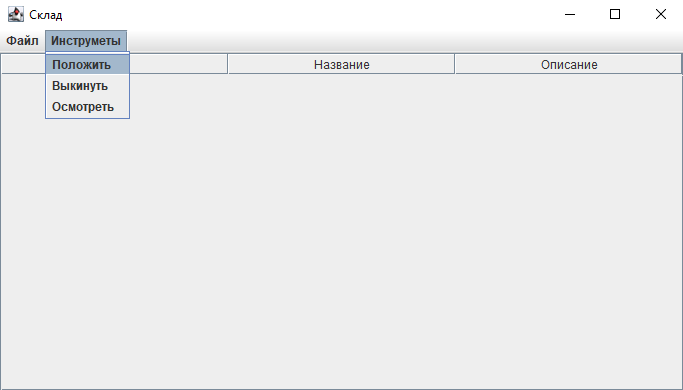


Рис. 2 – Выпадающее меню для взаимодействия с приложением



Рис. 3 – Выпадающее меню для добавления минерала

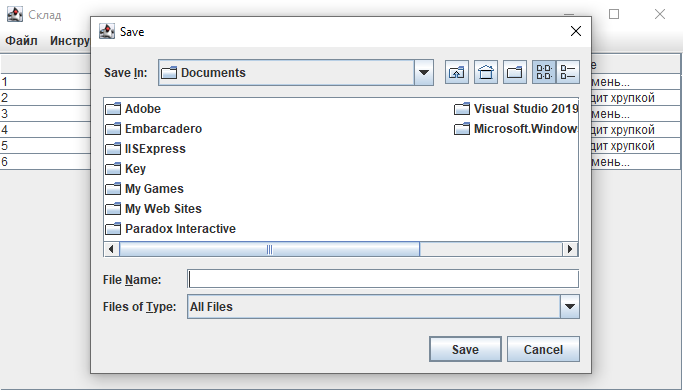


Рис. 4 – Диалоговое окно для сохранения файла

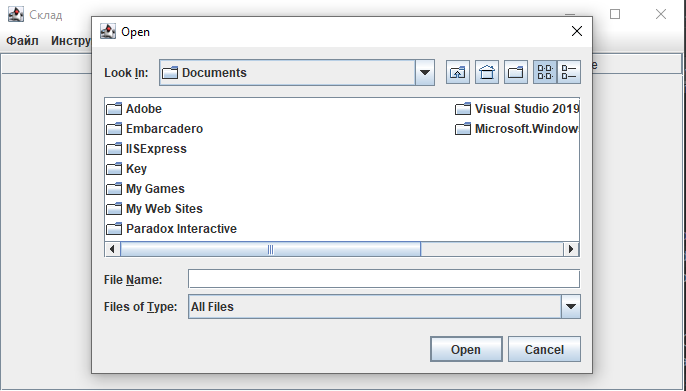


Рис. 5 – Диалоговое окно для открытия файла

На Рис 1 мы видим главную форму приложения, которая состоит из строки меню, таблицы. Для добавления новой строчки нужно нажать на кнопку «Инструменты», при этом выпадает меню Рис.2. Нажатие кнопки «Положить» выходит окно Рис 3. Тут пользователю нужно выбрать тип минерала, который необходимо добавить (Сыпучий минерал, Самоцвет или Камень).

Далее на панели меню есть вторая кнопка “Файл”, при ее нажатии выпадают кнопки “Сохранить файл” и “Открыть файл” при их нажатии выводится диалоговое окно выбора файла.

3 Описание структуры классов

Main.java – Запуск программы

MainWindow.java – Отвечает за вывод главного окна

TableMode.java – Отвечает за таблицу

SelectWindow.java – Отвечает за окно добавления, удаления, осмотра данных

Controller.java – Отвечает за обработку добавления, удаления, осмотра данных, а также сохранения и открытия файлов.

4 Таблица тестов

Таблица 1 - Тесты

|  |  |  |
| --- | --- | --- |
| № | Действия | Выходные данные |
| 1 | Нажимаем на кнопку «Инструменты» и нажимаем на кнопку «Положить» | Открывается окно добавления. |
| 2 | Добавляем «Сыпучий минерал» | Происходит добавление в таблицу |
| 3 | Добавляем «Самоцвет» | Происходит добавление в таблицу |
| 4 | Добавляем «Камень» | Происходит добавление в таблицу |
| 5 | Добавляем «Случайный минерал» | Происходит добавление в таблицу |
| 6 | Нажимаем на кнопку «Инструменты» и нажимаем на кнопку «Выкинуть» | Происходит удаление последней строки |
| 7 | Выделяем строку. Нажимаем на кнопку «Инструменты» и нажимаем на кнопку «Выкинуть» | Происходит удаление выделенной строки |
| 8 | Нажимаем на кнопку «Файл» и нажимаем на кнопку «Сохранить файл» | Выводится диалоговое окно сохранения в файл |
| 9 | Нажимаем на кнопку «Файл» и нажимаем на кнопку «Открыть файл» | Выводится диалоговое окно открытия файла |

5 Результаты тестирования

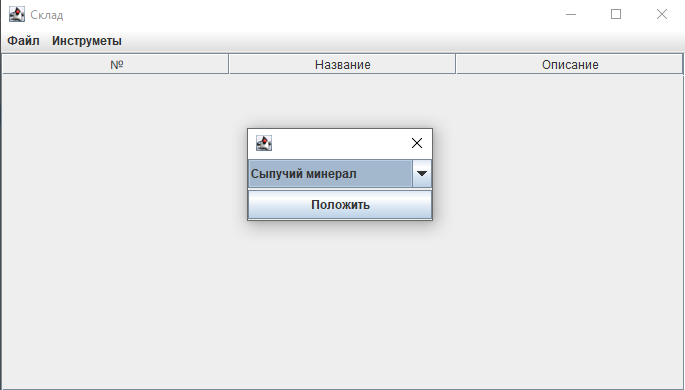


Рис. 6 – Тест 1

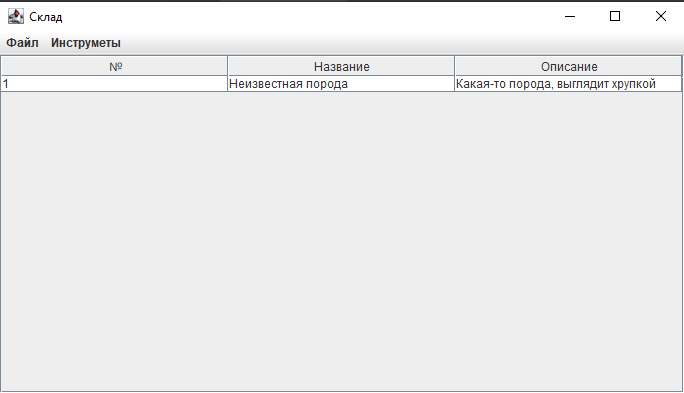


Рис. 7 – Тест 2

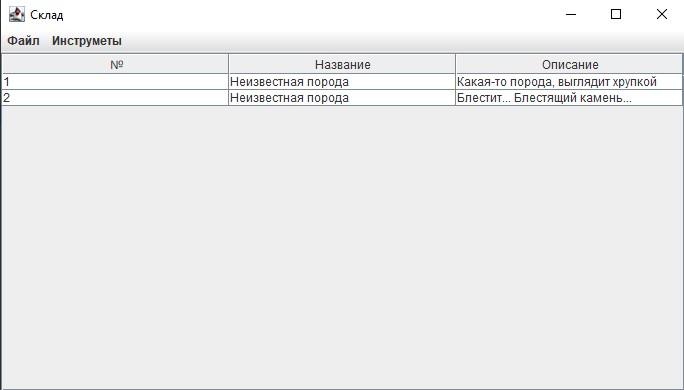


Рис. 8 – Тест 3

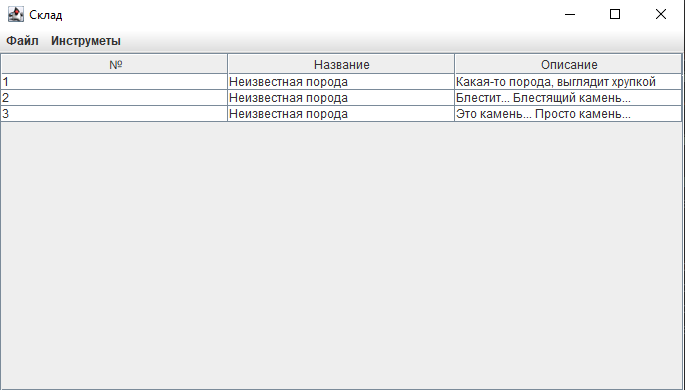


Рис. 9 – Тест 4

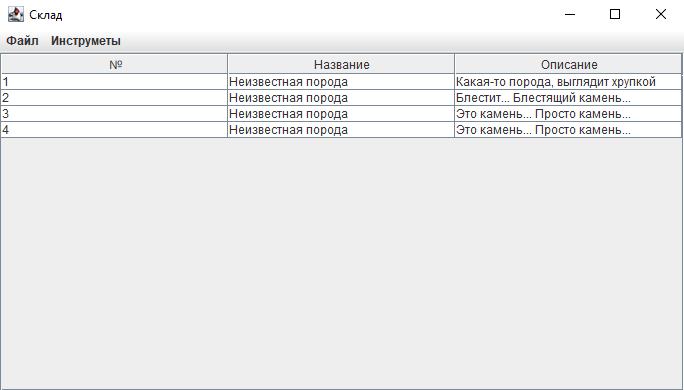


Рис. 10 – Тест 5

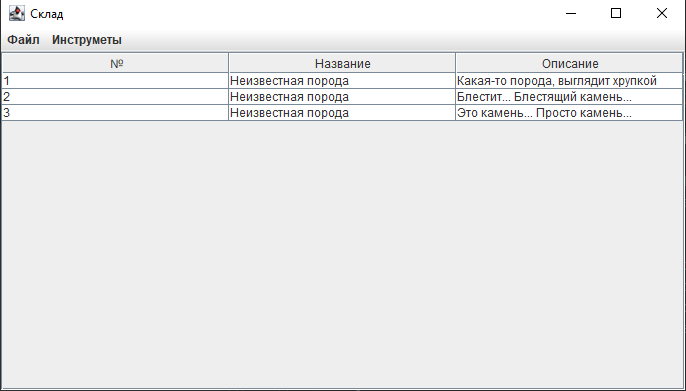
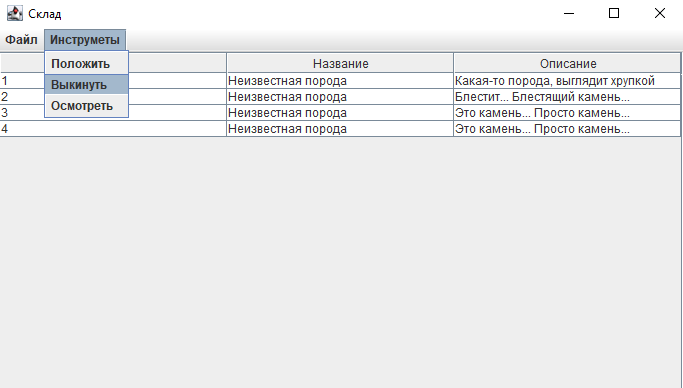


Рис. 11-12 – Тест 6

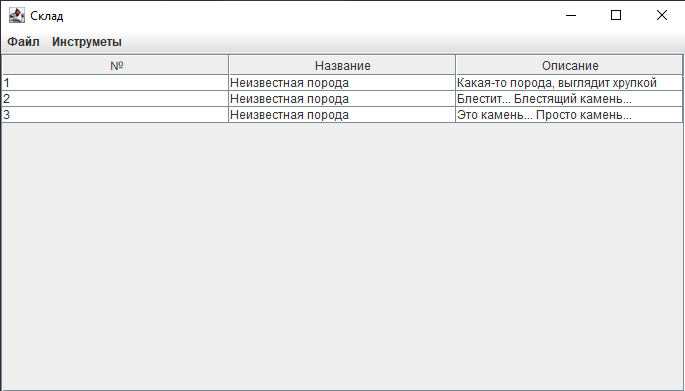
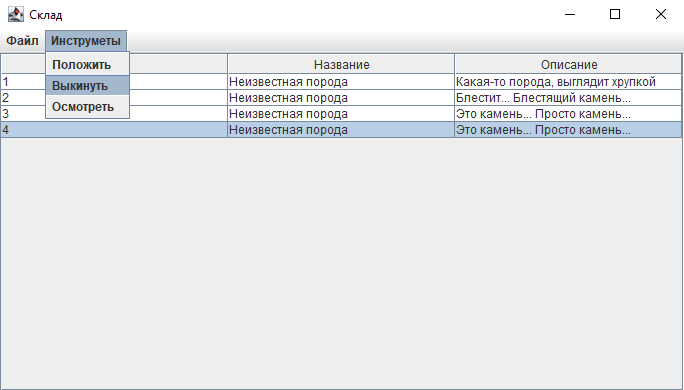


Рис. 13-14 – Тест 7

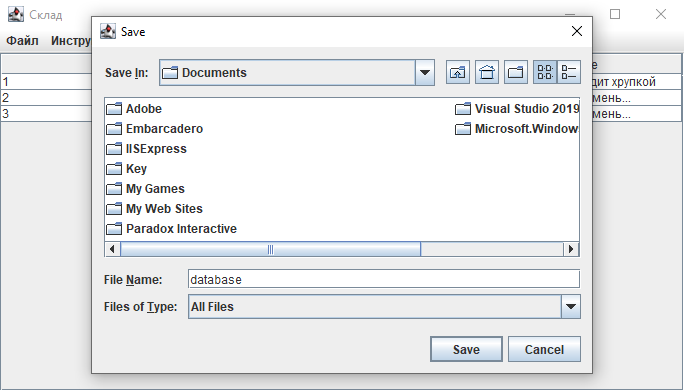


Рис. 15 – Тест 8

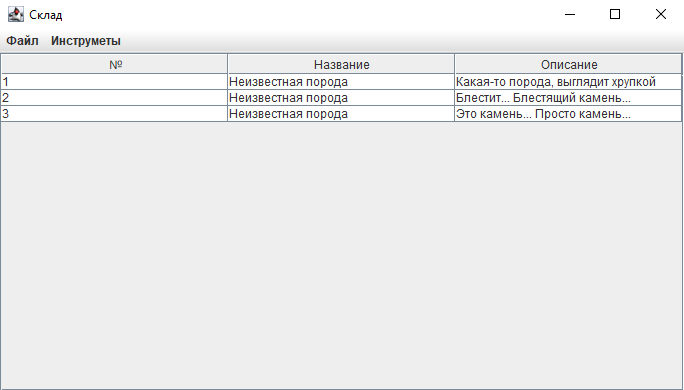
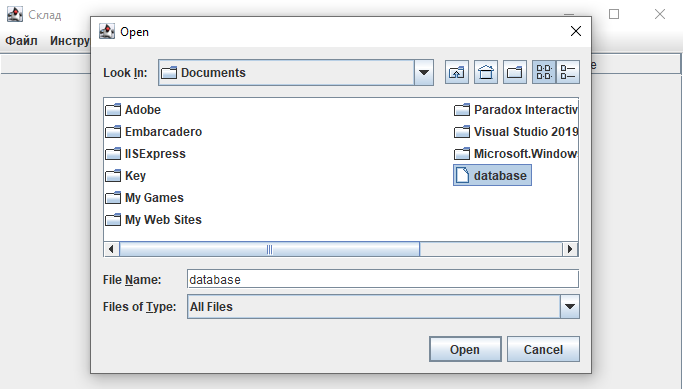


Рис. 16-17 – Тест 9

6 Листинг исходного кода

Класс Main

package Minerals;  
  
import Controllers.Controller;  
import Windows.MainFrame;  
  
public class Main {  
  
 public static void main(String[] args){  
 new Controller().execute(new MainFrame("Склад"));  
  
 }  
}

Класс MainFrame

package Windows;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.io.File;  
  
public class MainFrame extends JFrame{  
 public JMenuItem addMenuItem;  
 public JMenuItem deleteMenuItem;  
 public JMenuItem observeMenuItem;  
 public JMenuItem openFile;  
 public JMenuItem saveFile;  
 public TableMode tableMode;  
 public JTable table;  
 public SelectWindow selectWindow = new SelectWindow();  
 private JFileChooser chooser = new JFileChooser();  
  
 public MainFrame(String nameFrame){  
 super(nameFrame);  
 setSize(700,400);  
 setLayout(new GridLayout());  
  
 JMenuBar menuBar = new JMenuBar();  
 JMenu menuFile = new JMenu("Файл");  
 JMenu menu = new JMenu("Инструметы");  
  
 addMenuItem = new JMenuItem("Положить");  
 deleteMenuItem = new JMenuItem("Выкинуть");  
 observeMenuItem = new JMenuItem("Осмотреть");  
 openFile = new JMenuItem("Открыть");  
 saveFile = new JMenuItem("Сохранить");  
  
 menuFile.add(openFile);  
 menuFile.add(saveFile);  
 menu.add(addMenuItem);  
 menu.add(deleteMenuItem);  
 menu.add(observeMenuItem);  
  
 menuBar.add(menuFile);  
 menuBar.add(menu);  
 setJMenuBar(menuBar);  
  
 tableMode = new TableMode();  
 table = new JTable(tableMode);  
 JScrollPane pan = new JScrollPane(table);  
 add(pan);  
  
 setDefaultCloseOperation(*EXIT\_ON\_CLOSE*);  
 setVisible(true);  
 }  
  
 public File showDialog(boolean isSave) {  
 int option = isSave ? chooser.showSaveDialog(this) : chooser.showOpenDialog(this);  
 if(option == JFileChooser.*APPROVE\_OPTION*){  
 return chooser.getSelectedFile();  
 }  
 return null;  
 }  
  
 public void showError(String message){  
 JOptionPane.*showMessageDialog*(null, message);  
 }  
}

Класс TableMode

package Windows;  
  
import Minerals.Storage;  
  
import javax.swing.table.AbstractTableModel;  
  
public class TableMode extends AbstractTableModel {  
 public void update() {  
 fireTableDataChanged();  
 }  
  
 @Override  
 public int getRowCount() {  
 return Storage.*mineralList*.size();  
 }  
  
 @Override  
 public int getColumnCount() {  
 return 3;  
 }  
  
 @Override  
 public Object getValueAt(int rowIndex, int columnIndex) {  
 return switch (columnIndex) {  
 case 0 -> rowIndex + 1;  
 case 1 -> Storage.*mineralList*.get(rowIndex).nameMineral;  
 case 2 -> Storage.*mineralList*.get(rowIndex).description;  
 default -> throw new IllegalStateException("Unexpected value: " + columnIndex);  
 };  
 }  
  
 public String getColumnName(int columnIndex) {  
 return switch (columnIndex) {  
 case 0 -> "№";  
 case 1 -> "Название";  
 case 2 -> "Описание";  
 default -> throw new IllegalStateException("Unexpected value: " + columnIndex);  
 };  
 }  
}

Класс SelectWindow

package Windows;  
  
import javax.swing.\*;  
import java.awt.\*;  
  
public class SelectWindow extends JDialog {  
  
 public JButton button;  
 private String[] types = new String[]{"Сыпучий минерал","Самоцвет","Камень","Случайный"};  
 private JComboBox comboBox;  
  
 public SelectWindow(){  
 super();  
 JPanel panel = new JPanel();  
 panel.setLayout(new GridLayout(2,2,2,2));  
 comboBox = new JComboBox(types);  
 button = new JButton("Положить");  
  
 panel.add(comboBox);  
 panel.add(button);  
 add(panel);  
 pack();  
 setSize(200, 100);  
 }  
  
 public String getTypeChoice() {  
 return comboBox.getSelectedItem().toString();  
 }  
}

Класс Controller

package Controllers;  
  
import Minerals.\*;  
import Windows.MainFrame;  
import Windows.SelectWindow;  
  
import java.io.File;  
import java.io.FileWriter;  
import java.io.IOException;  
import java.nio.file.Files;  
import java.nio.file.Path;  
import java.util.ArrayList;  
  
public class Controller {  
 private MainFrame view;  
  
 private void executeSelect(SelectWindow select) {  
 select.button.addActionListener(e -> {  
 String type = view.selectWindow.getTypeChoice();  
  
 Mineral mineral = null;  
 switch(type) {  
 case "Сыпучий минерал" -> mineral = new Loose();  
 case "Самоцвет" -> mineral = new Gem();  
 case "Камень" -> mineral = new Stone();  
 case "Случайный" -> mineral = Storage.*randomMineral*();  
 }  
  
 Storage.*addMineral*(mineral);  
 view.tableMode.update();  
 select.dispose();  
 });  
 }  
  
 public void execute(MainFrame frame) {  
 view = frame;  
 executeSelect(view.selectWindow);  
  
  
 view.addMenuItem.addActionListener(e -> {  
 view.selectWindow.setVisible(true);  
 });  
  
 view.observeMenuItem.addActionListener(e ->{  
 try{  
 Mineral mineral = Storage.*mineralList*.get(view.table.getSelectedRow());  
 if (mineral instanceof Loose) {  
 view.showError("Рассыпалось");  
 }  
 mineral.observe();  
 }  
 catch (IndexOutOfBoundsException err) {  
 if (Storage.*mineralList*.size() == 0) return;  
 Storage.*mineralList*.get(Storage.*mineralList*.size() - 1).observe();  
 }  
 view.tableMode.update();  
 });  
  
 view.deleteMenuItem.addActionListener(e -> {  
 try {  
 Storage.*removeMineral*(view.table.getSelectedRow());  
 } catch (IndexOutOfBoundsException err) {  
 if (Storage.*mineralList*.size() == 0) return;  
 Storage.*removeMineral*(Storage.*mineralList*.size() - 1);  
 }  
 view.tableMode.update();  
 });  
  
 view.saveFile.addActionListener(e ->{  
 File file = view.showDialog(true);  
 if(file == null) return;  
 String data = constructor(Storage.*mineralList*);  
 try(FileWriter writer = new FileWriter(file)){  
 writer.write(data);  
 } catch (IOException ioException) {  
 view.showError("При записи файла произошла ошибка!");  
 }  
 });  
  
 view.openFile.addActionListener(e ->{  
 File file = view.showDialog(false);  
 if(file == null) return;  
 try {  
 Storage.*mineralList* = destructor(Files.*readString*(Path.*of*(String.*valueOf*(file))));  
 } catch (IOException ioException) {  
 view.showError("Ошибка в чтении файла");  
 }  
 catch (IndexOutOfBoundsException | IllegalArgumentException err){  
 view.showError("Неверный формат файла");  
 }  
 view.tableMode.update();  
 });  
 }  
  
 private String constructor(ArrayList<Mineral> mineralList){  
 StringBuilder sb = new StringBuilder();  
 for(Mineral mineral : mineralList){  
 sb.append(mineral.nameMineral).append(";").append(mineral.description).  
 append(";").append(mineral.isObserved).append(";").append(mineral.type).  
 append("\r\n");  
 }  
 return sb.toString();  
 }  
  
 private ArrayList<Mineral> destructor(String line) throws IndexOutOfBoundsException,IllegalArgumentException{  
 ArrayList<Mineral> arrayList = new ArrayList<>();  
 String[] stringArray = line.split("\r\n");  
 for(String mineralString : stringArray){  
 String[] stringArrayLine = mineralString.split(";");  
 String name = stringArrayLine[0];  
 String description = stringArrayLine[1];  
 boolean isObserve = Boolean.*parseBoolean*(stringArrayLine[2]);  
 String type = stringArrayLine[3];  
 Mineral mineral = switch (type){  
 case "Gem"-> new Gem(name,description,type,isObserve);  
 case "Loose"-> new Loose(name,description,type,isObserve);  
 case "Stone"-> new Stone(name,description,type,isObserve);  
 default -> throw new IllegalArgumentException("Неверный тип в файле");  
 };  
 arrayList.add(mineral);  
 }  
 return arrayList;  
 }  
}

Список использованных источников

1. Методические указания по выполнению лабораторных работ / сост.: В.Л. Аршинский. – Иркутск : Изд-во ИРНИТУ, 2017. – 24 c.
2. API Specification for the Java 7 SE. [официальный сайт] URL: <http://docs.oracle.com/javase/7/docs/api/> (дата обращения:08.09.2020)
3. The Java Tutorials. SE [электронный ресурс], URL:<http://docs.oracle.com/javase/tutorial/index.html> (дата обращения:08.09.2020)