# 1 Лабораторная работа №1 Основы программирования на языке Java в среде Eclipse

### 1.1. Цель работы

В ходе выполнения данной лабораторной работы необходимо ознакомиться с функциональными возможностями среды разработки Eclipse, основами языка Java, приобрести практические навыки создания консольных приложений на языке Java, с возможностью доступа к файлам.

#### 1.2. Постановка задачи

Был выдан вариант 13.

В соответствии с вариантом задания, представленным в п. 4 необходимо разработать программу на языке Java, выполняющую требуемые действия. Предусмотреть ввод входных данных с клавиатуры (по умолчанию) или из файла (при запуске с параметром -i <filename>). Предусмотреть вывод результатов на экран (по умолчанию) или в файл (при запуске с параметром -o <filename>). Предусмотреть возможность запуска с 2 параметрами: -i <filename1> -o <filename2>.Предусмотреть обработку ошибок с использованием операторов языка Java: try, catch и finally.

Ознакомившись со средствами отладки программ в среде Eclipse выполнить отладку разработанной программы.

Проверить правильность работы программы на тестовых примерах корректных и не корректных входных данных.

### 1.3. Ход выполнения работы

### 1.3.1. Текст программы

### Текст программы представлен в Листингах 1 — 5.

### Листинг 1 — Основной класс Арр

```
import java.io.FileNotFoundException;
import java.io.IOException;
class App
  public static void main (String[] args) throws IOException
    try
    {
      new UserInteraction(
        new CommandStringArguments(args).parse(),
        "Введите коэффициенты квадратного уравнения: ",
        "Ответ").readNumbers().writeOutput();
    catch (NumberFormatException e)
      System.out.println("Ошибка форматирования числа.");
    catch (IllegalArgumentException | FileNotFoundException e)
      System.out.println(e.getMessage());
    catch (IOException e)
      System.err.println("IO Error.");
 }
}
```

## Листинг 2 — Класс CommandStringArguments

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
class CommandStringArguments
  private final String[] args;
  public CommandStringArguments (String[] args)
    this.args = args;
  public Params parse () throws IllegalArgumentException, FileNotFoundException
    InputStream inStream = System.in;
    OutputStream outStream = System.out;
    boolean interactiveInput = true, interactiveOutput = true;
    int i = 0;
```

```
try
      for (; i < args.length; ++i)</pre>
        switch (args[i])
        case "-i", "--input":
         ++i:
          inStream = new FileInputStream(new File(args[i]));
          interactiveInput = false;
         break;
                  "--output":
        case "-o",
          ++i;
          outStream = new FileOutputStream(new File(args[i]));
          interactiveOutput = false;
          break;
        default:
          throw new IllegalArgumentException(String.format("[ERROR] Parametr %s is unknown.", args[i]));
     }
    catch (IndexOutOfBoundsException e)
      throw new IllegalArgumentException(String.format("Expected filename after %s param.", args[i - 1]));
    return new Params(
      new BufferedReader(new InputStreamReader(inStream)),
      new BufferedWriter(new OutputStreamWriter(outStream)),
      interactiveInput.
      interactiveOutput);
  }
}
       Листинг 3 — Класс Params
import java.io.BufferedReader;
import java.io.BufferedWriter;
class Params
  public final BufferedReader in;
  public final BufferedWriter out;
  public final boolean interactiveInput, interactiveOutput;
  public Params (BufferedReader in, BufferedWriter out, boolean interactiveInput, boolean interactiveOutput)
    this.in = in;
    this.out = out;
    this.interactiveInput = interactiveInput;
    this.interactiveOutput = interactiveOutput;
  }
}
       Листинг 4 — Класс Quadratic Equation
import java.util.List;
class QuadraticEquation
{
  private final double a, b, c;
  public QuadraticEquation (double a, double b, double c)
    this.a = a;
    this.b = b;
    this.c = c;
```

public List<Double> solve () throws ArithmeticException

```
{
    final double sqrt_D = Math.sqrt(b * b - 4 * a * c);
    if (sqrt_D == sqrt_D)
    {
        return List.of((-b + sqrt_D) / 2, (-b - sqrt_D) / 2);
    }
    else
    {
        throw new ArithmeticException("The discriminant is less than zero.");
    }
}
```

### Листинг 5 — Класс UserInteraction

```
import java.io.IOException;
import java.util.List;
class UserInteraction
{
  private final Params params;
  private final String inMessage;
  private final String outMessage;
  private List<Double> numbers;
  public UserInteraction (Params params, String inMessage, String outMessage)
    this.params = params;
    this.inMessage = inMessage;
    this.outMessage = outMessage;
  public UserInteraction readNumbers () throws IOException
    if (params.interactiveInput)
      System.out.print(inMessage + ": ");
    this.numbers = List.of(params.in.readLine().split("\\s")).stream().map(x -> Double.parseDouble(x)).toList();
    return this;
  public UserInteraction writeOutput () throws IOException
  {
    try
    {
      params.out
        .write(
          (params.interactiveOutput ? outMessage + ": " : "") + String
              new QuadraticEquation(numbers.get(0), numbers.get(1), numbers.get(2))
                .stream()
                .map(x -> x.toString())
                .toList()));
    catch (ArithmeticException e)
     params.out.write(e.getMessage() + "\n");
    catch (IndexOutOfBoundsException e)
      params.out.write("ERROR: Not enought params.\n");
    params.out.flush();
    return this;
 }
}
```

#### 1.3.2. Результаты тестирования

Было проведено тестирование программы. На рисунке 1 запущена программа, ей переданы три числа. В ответ программа вернула решение задачи.

Рисунок 1 — Запуск программы

На рисунке 2 предемонстрирован запуск программы. Ей было передано два числа и один символ. В результате была выведена ошибка преобразования слава в число.

Рисунок 2 — Запуск программы и ввод неверных параметров

На рисунке 3 предемонстрирован запуск программа с опцией -і. Файл существует. Программа вывела на экран верный результат.

Рисунок 3 — Запуск программы с опцией -і

На рисунке 4 предемонстрирован запуск программа с опцией -о. Программа записала в указанный файл верный результат.

Рисунок 4 — Запуск программы с опцией -о

### Вывод

При выполнении лабораторной работы были получены практические навыки написания программы на языке Java в среде разработки Eclipse. Были изучены предоставляемые языком средства чтения/записи в файл/из файла, создания и перехвата исключений. Получены навыки отладки программ в IDE Eclipse.