**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**

1.2.1 В соответствии с вариантом задания (Вариант – 11), необходимо разработать программу на языке Java, выполняющую требуемые действия: «Найти все положительные степени двойки, значение которых не превышает величины введенного с клавиатуры числа». 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.

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

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

**1.3 Ход работы**

Была написана программа на языке программирования Java согласно варианту:

import java.util.Arrays;

import java.util.Scanner; //для чтения всякого

import java.io.\*; //для работы с файлами

public class Main {

public static void main(String[] args) {

System.out.println("Программа находит все положительные степени двойки,значение которых не превышает величины введенного числа.");

int number = 0;

//Если в cmd были переданы аргументы вывести их на экран

if (args.length > 0) {

System.out.print("Аргументы коммандной строки на входе:");

System.out.println(Arrays.toString(args));

}

//Проверка на наличие аргументов коммандной строки

String inputFile = "";

String outputFile = "";

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

//если есть флаг -i то следующий аргумент коммандной строки это название input файла

if (args[i].equals("-i"))

inputFile = args[++i];

//если есть флаг -o то следующий аргумент коммандной строки это название output файла

else if (args[i].equals("-o"))

outputFile = args[++i];

}

//если есть флаг -i попытка открыть файл с входным значением

if (!inputFile.isEmpty()) {

try {

//для обращения к input файлу

BufferedReader fileIn = new BufferedReader(new FileReader(inputFile));

//превращение строки из файла в число

number = Integer.parseInt(fileIn.readLine().trim());

fileIn.close();

}

catch (IOException e) {

System.err.println("Ошибка при работе с input файлом " + e);

}

catch (NumberFormatException e) {

System.err.println("Ошибка преобразования входных данных " + e);

Scanner scan = new Scanner(System.in);

System.out.print("Введите число вручную >> ");

number = scan.nextInt();

}

}

else {

Scanner scan = new Scanner(System.in);

System.out.print("Введите число >> ");

number = scan.nextInt();

}

//если есть флаг -o попытка открыть файл для вывода результатов

if (!outputFile.isEmpty()) {

try {

//вывод в файл результатов работы программы

PrintWriter pw = new PrintWriter(outputFile);

pw.println("Программа находит все положительные степени двойки,значение которых не превышает величины введенного числа.");

pw.println("Введённое число >> " + number);

for (int i = 0; Math.pow(2, i) <= number; i++) {

pw.println("2^" + i + " = " + Math.pow(2, i));

}

pw.close();

System.out.println("Результаты работы программы выведены в файл");

} catch (IOException e) {

System.err.println("Ошибка при работе с output файлом " + e);

}

}

else {

System.out.println("Введённое число >> " + number);

for (int i = 0; Math.pow(2, i) <= number; i++) {

System.out.println("2^" + i + " = " + Math.pow(2, i));

}

}

}

}

Затем с помощью команды “javac Main.java” в командной строке был создан файл java.class.

Для запуска программы из командной строки и работы с параметрами командной строки использовалась команда “java Main «аргументы командной строки»”.

Была запущена программа без использования аргументов командной строки, результаты работы представлены на рисунке 1.1.

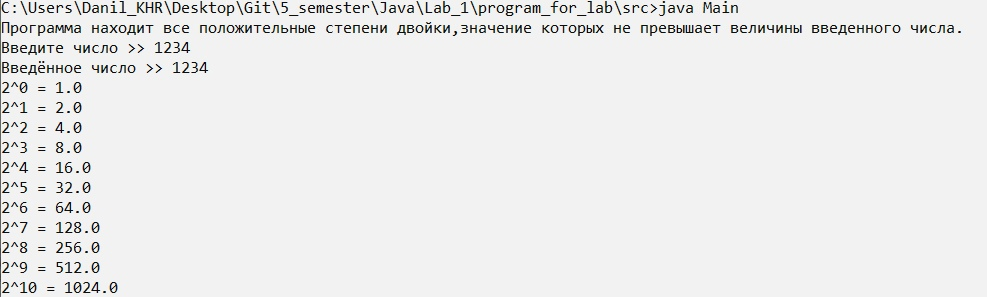


Рисунок 1.1 – Результат работы программы без аргументов командной строки

Так же были проделаны различные тестовые примеры работы программы с параметрами командной строки, отвечающими за входные значения из файла, за вывод результатов программы в файл, за оба предыдущих значения одновременно, а также с неверными значениями во входном файле. Результаты представлены на рисунках 1.2 – 1.7.

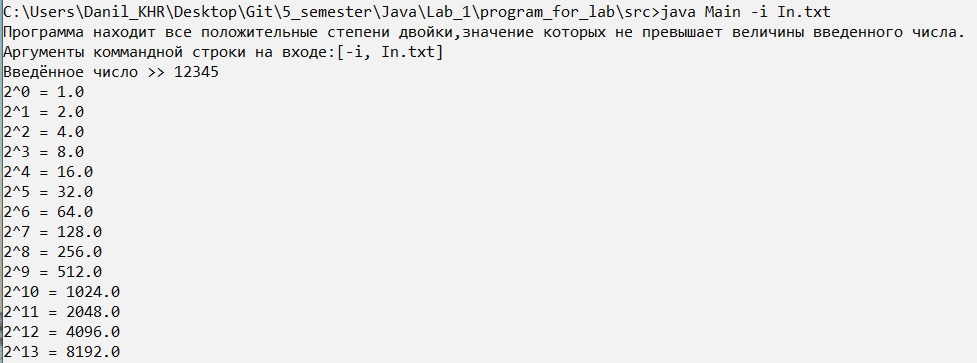


Рисунок 1.2 – Результат работы программы со входными данными из файла

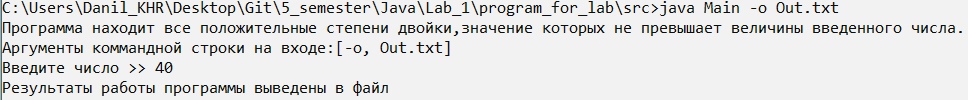


Рисунок 1.3 – Результат работы программы с выводом данных в файл

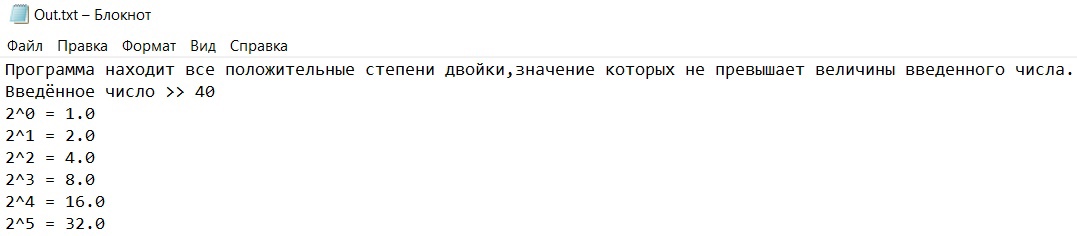


Рисунок 1.4 – Данные в файле в результате работы программы

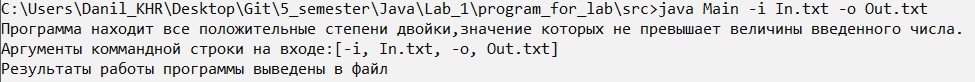


Рисунок 1.5 – Корректная работа программы при указании нескольких аргументов командной строки

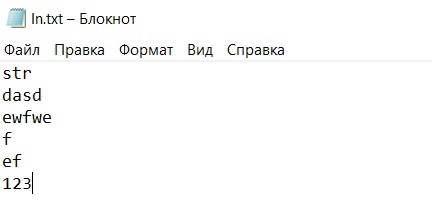


Рисунок 1.6 – Некорректные входные данные в файле

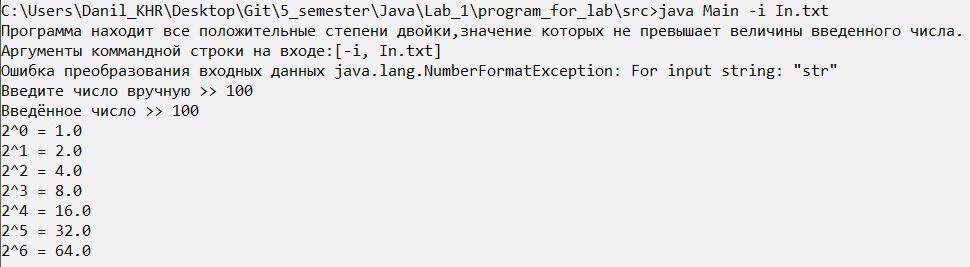


Рисунок 1.7 – Результат работы программы при некорректных входных данных

**Выводы**

В ходе выполнения данной лабораторной работы были изучены основы программирования на языке Java. Изучены потоки ввода/вывода и классы-обёртки над ними. Освоен механизм отлова ошибок try-catch-finally. Освоены навыки работы с программами на языке Java с использованием командной строки и её аргументов.