# ЛАБОРАТОРНАЯ РАБОТА №1

# ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ JAVA В СРЕДЕ ECLIPSE

Цель работы

Ознакомиться с функциональными возможностями среды разработки Eclipse, основами языка Java, приобрести практические навыки создания консольных приложений на языке Java с возможностью доступа к файлам.

Задачи

1. В соответствии с вариантом задания 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*;
2. Ознакомившись со средствами отладки программ в среде Eclipse, выполнить отладку разработанной программы;
3. Проверить правильность работы программы на тестовых примерах, корректных и не корректных входных данных;

Вариант задания

Вариант 13: написать программу решения квадратного уравнения по заданным коэффициентам *a*, *b* и *c*.

Ход работы

В среде разработки Eclipse была написана программа, осуществляющая обработку параметров, введённых в командной строке, и вычисляющая корни квадратного уравнения по считанным параметрам.

Программа была скомпилирована через компилятор JDK (javac). Затем она была протестирована для ввода и вывода данных вручную (без параметров). Программа справилась с задачей верно (Рисунок 1).

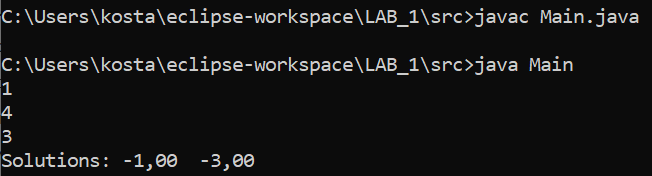


Рисунок 1 – Тестирование ручного ввода-вывода

Программа была протестирована для параметров *-i <filename>*. Для этого был создан файл input.txt, содержащий параметры квадратного уравнения (Рисунок 2). При запуске программа верно считала данные и вывела правильный ответ (Рисунок 3).

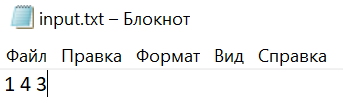


Рисунок 2 – Файл для считывания данных



Рисунок 3 – Тестирование ввода из файла

Программа была протестирована для параметров *-o <filename>* (Рисунок 4). В результате программой был создан файл output.txt, в который она вывела правильный ответ (Рисунок 5).

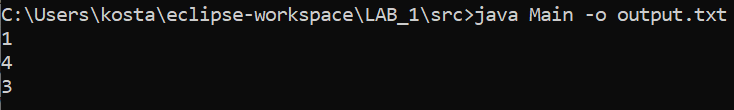


Рисунок 4 – Тестирование вывода в файл

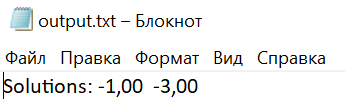


Рисунок 5 – Файл вывода данных

Также было подтверждено, что программа может одновременно распознавать как файл для ввода, так и файл для вывода.

При попытке ввода букв вместо чисел программа ошибку (Рисунок 6).

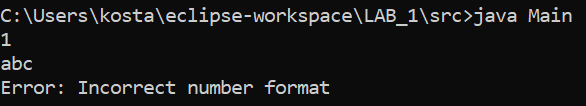


Рисунок 6 – Сообщение об ошибке считывания числа

При попытке ввода из несуществующего файла программа вывела ошибку (Рисунок 7).



Рисунок 7 – Ошибка нахождения файла

При параметрах, дающих дискриминант меньше 0, программа вывела сообщение об ошибке (Рисунок 8).

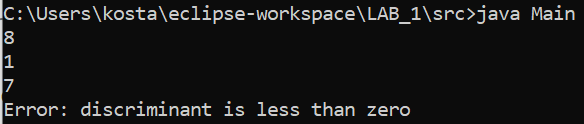


Рисунок 8 – Ошибка отрицательного дискриминанта

При указании параметра *-i* без *<filename>* программа вывела ошибку (Рисунок 9).



Рисунок 9 – Ошибка недостающего параметра

На основании проведённых исследований было заключено, что программа верно справляется с поставленными ей задачами.

Текст программы

**import** java.io.FileInputStream;

**import** java.io.PrintStream;

**import** java.util.Scanner;

**import** java.util.InputMismatchException;

**import** java.io.IOException;

**import** java.io.FileNotFoundException;

**public** **class** Main {

**public** **static** **void** main(String[] args)

{

**int** i = 0;

**try**

{

**for** (; i < args.length; ++i) // перебор параметров с консоли

{

**switch** (args[i])

{

**case** "-i": // перенаправление ввода из файла

i++;

System.*setIn*(**new** FileInputStream(args[i]));

**break**;

**case** "-o": // перенаправление вывода в файл

i++;

System.*setOut*(**new** PrintStream(args[i]));

**break**;

**default**: // ошибка параметра

**throw** **new** IllegalArgumentException();

}

}

// Считывание переменных a, b, c

Scanner scanner = **new** Scanner(System.***in***);

**int** a = scanner.nextInt();

**int** b = scanner.nextInt();

**int** c = scanner.nextInt();

scanner.close();

// Дискриминант и корни квадратного уравнения

**double** D = b \* b - 4 \* a \* c;

**if** (D < 0) { **throw** **new** ArithmeticException("Error: discriminant is less than zero"); }

**double** x1 = (-b + Math.*sqrt*(D)) / (2\*a);

**double** x2 = (-b - Math.*sqrt*(D)) / (2\*a);

// Вывод результата

**if** (x1 != x2) { System.***out***.println(String.*format*("Solutions: %.2f %.2f", x1, x2));}

**else** {System.***out***.println(String.*format*("Solution: %.2f", x1));}

}

**catch** (FileNotFoundException e) // файл не найден

{

System.***out***.printf("Error: Can't find '%s' file", args[i]);

}

**catch** (IndexOutOfBoundsException e) // недостающий параметр

{

System.***out***.printf("Error: Expected filename after %s param", args[i - 1]);

}

**catch** (IllegalArgumentException e) // неподдерживаемый параметр

{

System.***out***.printf("Error: Parametr %s is unknown", args[i]);

}

**catch** (IOException e) // ошибка ввода-вывода

{

System.***out***.println("Error: IO exception");

}

**catch** (InputMismatchException e) // неверная запись числа

{

System.***out***.println("Error: Incorrect number format");

}

**catch** (ArithmeticException e) // вычислительная ошибка

{

System.***out***.println(e.getMessage());

}

}

}

Вывод

В ходе работы было проведено ознакомление с функциональными возможностями среды разработки Eclipse и основами языка Java. Среди рассмотренных тем: переменные, идентификаторы, типы данных, операции, массивы, использование аргументов командной строки, потоки ввода-вывода, обработка исключений.

Результатом работы стала программа, написанная на языке Java в среде Eclipse, которая реализует обработку параметров командной строки, файловый и ручной ввод-вывод данных, математические операции и обработку исключений.

.