МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

федеральное государственное бюджетное образовательное учреждение высшего образования

**«Вологодский государственный университет»**

**Институт математики, естественных и компьютерных наук**

**Информатика и вычислительная техника**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4**

Создание классов на языке Java.

Дисциплина: «Объектно-ориентированное программирование»

Направление подготовки: 09.03.01. Информатика и вычислительная техника

|  |  |
| --- | --- |
| Руководитель | Сорокин А.Н. |
| Выполнили студенты | Пчелкина О.С. |
| Группа, курс | ВМ-41 |
| Дата сдачи | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Дата защиты | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись преподавателя)* |

Вологда

2022 г.

**Цель работы**: научиться разрабатывать классы на языке Java.

# Вариант 1

1. Создать базовый класс «уравнение» (абстрактный класс или интерфейс) с виртуальными функциями: вычисление корней, ввод уравнения с клавиатуры и вывод уравнения на экран.

Создать производные классы: класс линейных уравнений, класс квадратных уравнений.

В производных классах определить: скрытые поля, конструктор по умолчанию, конструктор с параметрами, методы для установки и получения значение полей, методы для вычисления корней уравнения, ввода и вывода уравнения.

2. Разработать программу, которая демонстрировала бы работоспособность всех методов классов. Для проверки определить коллекцию ссылок на базовый класс, которым присваиваются адреса различных объектов.

3. Разработать программу, которая демонстрировала бы возможности сериализации. Для проверки введите данные с клавиатуры, сохраните информацию в файл (требуется сохранить несколько объектов разных классов из задания 1), затем прочитайте данные в цикле и выведите информацию на экран (в этом цикле неизвестно количество и порядок объектов в файле).

Код программы:

import java.io.\*;

import java.util.Scanner;

interface Equation {

public void calculation();

public void input();

public void print();

}

class LinearEquation implements Equation, Serializable {

double linearA;

double linearB;

//конструктор по умолчанию

public LinearEquation() {}

//конструктор с параметрами

public LinearEquation(double linearA, double linearB) {

this.linearA = linearA;

this.linearB = linearB;

}

public void setLinearA(double linearA) {

this.linearA = linearA;

}

public void setB(double linearB) {

this.linearB = linearB;

}

public double getLinearA() {

return linearA;

}

public double getB() {

return linearB;

}

@Override

public void calculation() {

System.out.println("решение: корень уравнения х = " + -linearB / linearA);

}

@Override

public void input() {

Scanner in = new Scanner(System.in);

System.out.println("Введите аргумент а: ");

linearA = in.nextDouble();

System.out.println("Введите аргумент b: ");

linearB = in.nextDouble();

}

@Override

public void print() {

//тернарное выражение для того, чтобы ставить перед аргументом знак "+" в уравнении, если он положительный

System.out.printf("Линейное уравнение "

+ "%sx" + (linearB > 0 ? "+" : "")

+ "%s; ", linearA, linearB);

}

}

class QuadraticEquation implements Equation, Serializable {

double quadraticA;

double quadraticB;

double quadraticC;

public QuadraticEquation() {}

public QuadraticEquation(double quadraticA, double quadraticB, double quadraticC) {

this.quadraticA = quadraticA;

this.quadraticB = quadraticB;

this.quadraticC = quadraticC;

}

@Override

public void calculation() {

double D = Math.pow(quadraticB, 2) - 4 \* quadraticA \* quadraticC;

if (D > 0) {

double x1, x2;

x1 = (-quadraticB - Math.sqrt(D)) / (2 \* quadraticA);

x2 = (-quadraticB + Math.sqrt(D)) / (2 \* quadraticA);

System.out.println("корни уравнения: x1 = " + x1 + ", x2 = " + x2);

} else if (D == 0) {

double x;

x = -quadraticB / (2 \* quadraticA);

System.out.println("уравнение имеет единственный корень: x = " + x);

} else {

System.out.println("уравнение не имеет корней");

}

}

@Override

public void input() {

Scanner in = new Scanner(System.in);

System.out.println("Введите аргумент а: ");

quadraticA = in.nextDouble();

System.out.println("Введите аргумент b: ");

quadraticB = in.nextDouble();

System.out.println("Введите аргумент c: ");

quadraticC = in.nextDouble();

}

@Override

public void print() {

System.out.printf("Квадратное уравнение "

+ "%sx^2" + (quadraticB > 0 ? "+" : "")

+ "%sx" + (quadraticC > 0 ? "+" : "")

+ "%s; ", quadraticA, quadraticB, quadraticC);

}

}

Программа, демонстрирующая работу класса:

public class Main {

public static void main(String[] args) throws IOException {

//для линейного уравнения

System.out.println("Линейное уравнение (конструктор с параметрами):");

LinearEquation firstLinearEquation = new LinearEquation(2, 4);

firstLinearEquation.print();

firstLinearEquation.calculation();

System.out.println("Линейное уравнение (конструктор по умолчанию):");

LinearEquation secondLinearEquation = new LinearEquation();

secondLinearEquation.input();

secondLinearEquation.print();

secondLinearEquation.calculation();

System.out.println();

//для квадратного уравнения

System.out.println("Квадратное уравнение (конструктор с параметрами):");

QuadraticEquation firstQuadraticEquation = new QuadraticEquation(2, -6, 2);

firstQuadraticEquation.print();

firstQuadraticEquation.calculation();

System.out.println("Квадратное уравнение (конструктор по умолчанию):");

QuadraticEquation secondQuadraticEquation = new QuadraticEquation();

secondQuadraticEquation.input();

secondQuadraticEquation.print();

secondQuadraticEquation.calculation();

System.out.println();

//Проверка наследования

System.out.println("Проверка наследования");

//создаем массив объектов

Equation[] array = new Equation[2];

array[0] = firstLinearEquation;

array[1] = firstQuadraticEquation;

//выводим их

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

array[i].print();

array[i].calculation();

}

System.out.println();

System.out.println("Работа с файлом");

//создаем поток для записи объектов в файл

FileOutputStream fos = new FileOutputStream("D:\\lab4\\src\\main\\java\\test.txt");

//так как мы можем записывать только байты, то

//создаем поток для преобразования объекта в байты

ObjectOutputStream oos = new ObjectOutputStream(fos);

//записываем

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

oos.writeObject(array[i]);

}

//очищаем модуль записи

oos.flush();

oos.close();

//создаем потоки для вывода из файла и преобразования байтов обратно в объекты

FileInputStream fis = new FileInputStream("D:\\lab4\\src\\main\\java\\test.txt");

ObjectInputStream oin = new ObjectInputStream(fis);

//считываем и выводим

try{

while (true){

Equation eq = (Equation) oin.readObject();

eq.print();

System.out.println();

}

}catch (Exception e){

System.out.println("Конец вывода");

}

}

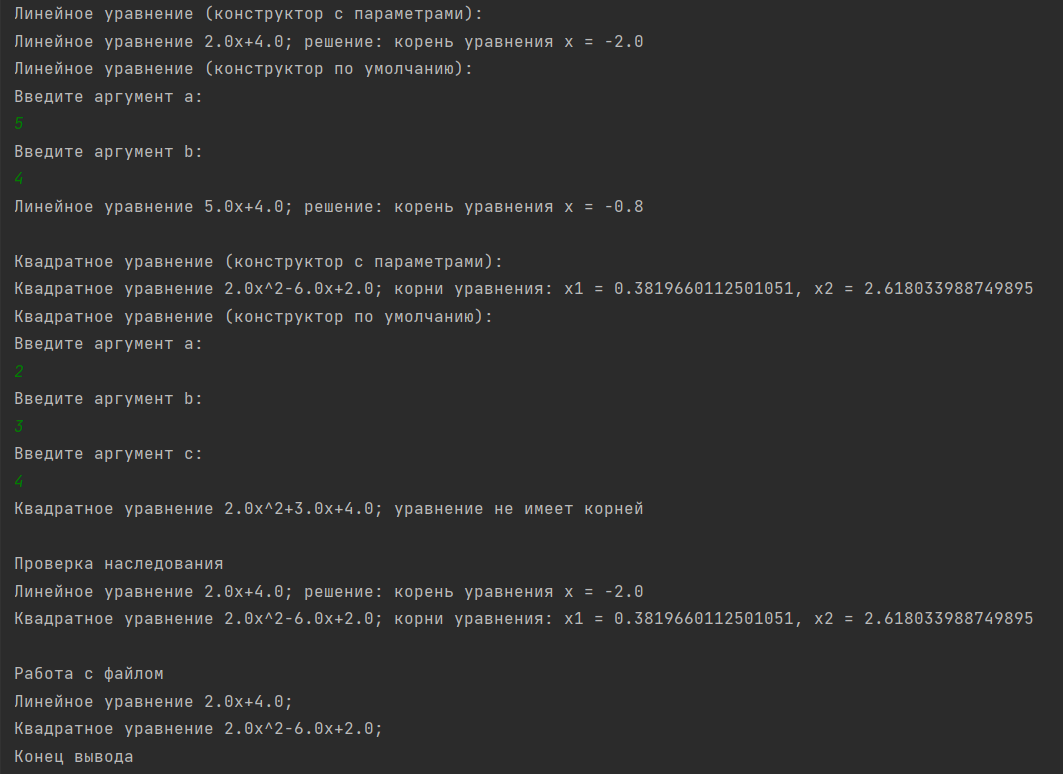


Рис. 1 – Работа программы

# Контрольные вопросы

1. Что понимается под термином «абстрактный класс»? В каких случаях применяются абстрактные классы?

*Абстрактным классом в Java называется класс, который содержит абстрактные методы — методы, пред объявлением которых указывается ключевое слово abstract и определение которых дается только в подклассах класса.*

*Абстрактные классы используются в качестве обобщенных концепций, на основе которых можно создавать более конкретные производные классы*

2. Что понимается под термином «интерфейс»? В каких случаях применяются интерфейсы?

*Интерфе́йс — программная/синтаксическая структура, определяющая отношение между объектами, которые разделяют определённое поведенческое множество и не связаны никак иначе. При проектировании классов, разработка интерфейса тождественна разработке спецификации.*

*Использование интерфейсов — один из вариантов обеспечения полиморфизма в объектных языках и средах. Все классы, реализующие один и тот же интерфейс, с точки зрения определяемого ими поведения, ведут себя внешне одинаково. Это позволяет писать обобщённые алгоритмы обработки данных, использующие в качестве типов параметры интерфейсов, и применять их к объектам различных типов, всякий раз получая требуемый результат.*