|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6) \_\_\_\_\_\_

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/12 Интеллектуальный анализ больших**

**данных в системах поддержки принятия решений**

**Отчет**

**по лабораторной работе 5**

**Название:** Исключения, файлы

**Дисциплина:** Языки программирования для работы с большими данными

Студент гр. ИУ6-23М **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_** И. Г. Калинин**\_\_**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_** П. В. Степанов **\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2023

**Цель лабораторной работы**

Познакомиться с базовыми понятиями языка Java, изучить механизмы создания собственных исключений. Изучить методы работы с файлами, решить набор задач согласно индивидуальному варианту.

**Фрагмент программного кода**

В качестве демонстрации результата приведен один из листингов проекта. Код на языке Java представлен в листинге 1.

Листинг 1 – Код на языке Java

|  |
| --- |
| import java.util.Arrays;  public class QuadraticEquation {  private double a;  private double b;  private double c;  public QuadraticEquation() {  this.a = 0;  this.b = 0;  this.c = 0;  }  public QuadraticEquation(double a, double b, double c) {  this.a = a;  this.b = b;  this.c = c;  }  public double[] findRoots() {  double discriminant = Math.pow(b, 2) - 4 \* a \* c;  try {  if (discriminant > 0) {  double root1 = (-b + Math.sqrt(discriminant)) / (2 \* a);  double root2 = (-b - Math.sqrt(discriminant)) / (2 \* a);  return new double[]{root1, root2};  } else if (discriminant == 0) {  double root = -b / (2 \* a);  return new double[]{root};  } else {  // Генерирование исключения ArithmeticException  throw new ArithmeticException("Discriminant is negative");  }  } catch (ArithmeticException e) {  // Обработка ошибки деления на ноль или других арифметических ошибок  System.err.println("Error finding roots: " + e.getMessage());  return new double[]{};  }  }  public double[] findExtremes() {  try {  double x = -b / (2 \* a);  double y = a \* Math.pow(x, 2) + b \* x + c;  return new double[]{x, y};  } catch (ArithmeticException e) {  // Обработка ошибки деления на ноль или других арифметических ошибок  System.err.println("Error finding extremes: " + e.getMessage());  return new double[]{};  }  }  public boolean isIncreasing() {  return a > 0;  }  public boolean isDecreasing() {  return a < 0;  }  public static void main(String[] args) {  QuadraticEquation[] equations = new QuadraticEquation[3];  equations[0] = new QuadraticEquation(); // Использование конструктора без параметров  equations[1] = new QuadraticEquation(1, -3, 2);  equations[2] = new QuadraticEquation(2, 1, 6);  double smallestRoot = Double.POSITIVE\_INFINITY;  double largestRoot = Double.NEGATIVE\_INFINITY;  for (QuadraticEquation equation : equations) {  double[] roots;  try {  roots = equation.findRoots();  for (double root : roots) {  if (root < smallestRoot) {  smallestRoot = root;  }  if (root > largestRoot) {  largestRoot = root;  }  }  } catch (Exception e) {  // Обработка исключений, возникающих при выполнении findRoots()  System.err.println("Error finding roots: " + e.getMessage());  }  }  System.out.println("Smallest root: " + smallestRoot);  System.out.println("Largest root: " + largestRoot);  }  } |

**Вывод**

В рамках данной лабораторной работы были реализованы все задачи, выданные преподавателем, а также изучены и выполнены все пункты целей данной лабораторной работы.