|  |  |
| --- | --- |
| 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 09.04.01/07 **Интеллектуальные системы анализа, обработки и интерпретации больших данных**

**Отчет**

|  |
| --- |
| **по лабораторной работе № 5** |

**Название:**

Исключения

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-23М |  |  | Т.А. Малкина |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

Вариант 1:

Выполнить задания на основе варианта 1 лабораторной работы 3, контролируя состояние потоков ввода/вывода. При возникновении ошибок, связанных с корректностью выполнения математических операций, генерировать и обрабатывать исключительные ситуации. Предусмотреть обработку исключений, возникающих при нехватке памяти, отсутствии требуемой записи (объекта) в файле, недопустимом значении поля и т.д.

Решение:

|  |
| --- |
| package com.company; import java.util.\*; import com.company.MyVector;   public class Main {  public static void main(String[] args) {  Scanner in = new Scanner(System.in);  /\*  Вариант 1  1. Определить класс Вектор размерности n.  Реализовать методы сложения, вычитания, умножения, инкремента, декремента, индексирования.  Определить массив из m объектов.  Каждую из пар векторов передать в методы, возвращающие их скалярное произведение и длины.  Вычислить и вывести углы между векторами.  2. Определить класс Вектор размерности n.  Определить несколько конструкторов.  Реализовать меткалярного произведеноды для вычисления модуля вектора, сия, сложения, вычитания, умножения на константу.  Объявить массив объектов.  Написать метод, который для заданной пары векторов будет определять, являются ли они коллинеарными или ортогональными.  \*/  System.out.println("------------------------------");  System.out.println("Лаб 3. Вариант 1,2");  System.out.println("Введите количество m объектов");  int m = in.nextInt();  MyVector[] masvec = new MyVector[m];  for (int i = 0; i < m; i++) {  masvec[i] = new MyVector(i,i);  }  //Вывод сложения, вычитания, умножения, инкремента, декремента  for (int i = 0; i < (m-1); i++) {  System.out.println("Операции сложения, вычитания, умножения для векторов "+i+" и "+(i+1)+" :");  masvec[i].oper(masvec[i+1]);  masvec[i].cos(masvec[i+1]);  masvec[i].kolort(masvec[i+1]);  }  for (int i = 0; i < m; i++) {  System.out.println("Операции инкремента, декремента для вектора "+i);  masvec[i].incdec();  masvec[i].len();  }  System.out.println("------------------------------");  System.out.println("Лаб 5. Вариант 1");  /\*  Вариант 1  Выполнить задания на основе варианта 1 лабораторной работы 3,  контролируя состояние потоков ввода/вывода.  При возникновении ошибок, связанных с корректностью выполнения математических операций,  генерировать и обрабатывать исключительные ситуации.  Предусмотреть обработку исключений, возникающих при нехватке памяти,  отсутствии требуемой записи (объекта) в файле, недопустимом значении поля и т.д.  \*/   try {  for (int i = 0; i < m; i++) {  masvec[i] = new MyVector(i,i);  }  }  catch (ArrayIndexOutOfBoundsException e){  System.out.println("Возникли ошибки при создании массива");   }  //Вывод сложения, вычитания, умножения, инкремента, декремента  for (int i = 0; i < (m-1); i++) {  System.out.println("Операции сложения, вычитания, умножения для векторов "+i+" и "+(i+1)+" :");  masvec[i].oper(masvec[i+1]);  try {  masvec[i].cos(masvec[i+1]);  }  catch (ArithmeticException e){  System.out.println("ArithmeticException");  }  try {  masvec[i].kolort(masvec[i+1]);  }  finally {  System.out.println("Не удалось определить оллинеарность и ортогональность");  }  }  for (int i = 0; i < m; i++) {  System.out.println("Операции инкремента, декремента для вектора "+i);  masvec[i].incdec();  masvec[i].len();  }      } }  package com.company;  public class MyVector {  int x,y;   public MyVector(int x, int y) {  this.x = x;  this.y = y;  }   public MyVector() {  this.x = (int) Math.random();  this.y = (int) Math.random();  }   public int getX() {  return x;  }   public int getY() {  return y;  }   void oper(MyVector v2){  System.out.println("Сумма. Вектор С с координатами: ["+(x+ v2.getX())+","+(y+v2.getY())+"]");  System.out.println("Разность. Вектор С с координатами: ["+(x- v2.getX())+","+(y-v2.getY())+"]");  System.out.println("Умножение. Скалярное произведение векторов: "+(x\*v2.getX()+y\*v2.getY()));  }   void incdec (){  System.out.println("Инкремент - ["+(x+1)+","+(y+1)+"]");  System.out.println("Декремент - ["+(x-1)+","+(y-1)+"]");  System.out.println("Умножение вектора на число - ["+(x\*2)+","+(y\*2)+"]");  }  void len (){  System.out.println("Длина(модуль) вектора - "+Math.sqrt(x\*x+y\*y));  }  void cos (MyVector v2){  double up,down,cos;  up = x\*v2.getX()+y\*v2.getY();  down = Math.sqrt(x\*x+y\*y)\*Math.sqrt(v2.getX()\*v2.getX()+ v2.getY()\* v2.getY());  cos = up/down;  System.out.println("Угол между векторами - "+cos);  }  void kolort (MyVector v2){  System.out.println("Коллинеарность и ортогональность: ");  double up,down,cos;  int skalar = x\*v2.getX()+y\*v2.getY();  if (skalar < 0.01){  System.out.println("Вектора коллинеарны");  }  else  System.out.println("Вектора не коллинеарны ");  up = x\*v2.getX()+y\*v2.getY();  down = Math.sqrt(x\*x+y\*y)\*Math.sqrt(v2.getX()\*v2.getX()+ v2.getY()\* v2.getY());  cos = up/down;  if (cos == 0){  System.out.println("Вектора ортогональны");  }  else  System.out.println("Вектора не ортогональны");  }  } |

Вариант 2:

Выполнить задания из варианта 2 лабораторной работы 3, реализуя собственные обработчики исключений и исключения ввода/вывода.

Решение:

|  |
| --- |
| package com.company; import java.util.\*; import com.company.Student; import com.company.Customer;   public class Main {   public static void main(String[] args) {  Scanner in = new Scanner(System.in);  /\*  Вариант 2  Создать классы, спецификации которых приведены ниже. Определить конструкторы и методы setТип(), getТип(), toString().  Определить дополнительно методы в классе, создающем массив объектов. Задать критерий выбора данных и вывести эти данные на консоль.  1. Student: id, Фамилия, Имя, Отчество, Дата рождения, Адрес, Телефон, Факультет, Курс, Группа. Создать массив объектов.  Вывести: a) список студентов заданного факультета; b) списки студентов для каждого факультета и курса;  c) список студентов, родившихся после заданного года; d) список учебной группы.  2. Customer: id, Фамилия, Имя, Отчество, Адрес, Номер кредитной карточки, Номер банковского счета. Создать массив объектов.  Вывести: a) список покупателей в алфавитном порядке; b) список покупателей, у которых номер кредитной карточки находится в заданном интервале.  \*/  System.out.println("------------------------------");  System.out.println("Вариант 2.1");  System.out.print("Введите количество студентов - ");  int n = in.nextInt();  try {  if (n <= 0) {  throw new MyExc(); // Вызов метода создания объекта MyException  }  ArrayList<Student> students = new ArrayList<Student>();  for (int i = 0; i < n; i++) {  students.add(new Student(i));  }  System.out.println("Студенты 1-го факультета:");  for (Student s : students){  if(s.getFaculty().equals("1")){  System.out.println(s.toString());  }  }  System.out.println("Список студентов факультета 1 курса 1:");  for(Student s : students){  if(s.getFaculty().equals("1")&s.getCourse().equals("1")){  System.out.println(s.toString());}  }  } catch (MyExc e) {  new MyExc().MyExc();  System.err.println("Error!");  }  System.out.println("------------------------------");  System.out.println("Вариант 2.2");  System.out.print("Введите количество Customer - ");  int n2 = in.nextInt();   try {  if (n2 <= 0) {  throw new MyExc(); // Вызов метода создания объекта MyException  }  ArrayList<Customer> customers= new ArrayList<Customer>();  for (int i = 0; i < n2; i++) {  customers.add(new Customer(i));  }  System.out.println("a) список покупателей в алфавитном порядке:");  Collections.reverse(customers);  System.out.println(customers);  System.out.println("b) список покупателей, у которых номер кредитной карточки находится в заданном интервале:");  for (Customer c : customers) {  if (c.getNum\_card() >= 1 && c.getNum\_card() <= 3){  System.out.println(c.toString());}  }  } catch (MyExc e) {  new MyExc().MyExc();  System.err.println("Error!");  }       /\*  Вариант 2 Выполнить задания из варианта 2 лабораторной работы 3, реализуя собственные обработчики исключений и исключения ввода/вывода.  \*/  Student excStudent = new Student(123);  try {  if (excStudent.id != 0) {  throw new MyExc(); // Вызов метода создания объекта MyException  }  } catch (MyExc e) {  new MyExc().MyExc();  System.err.println("Error!");  }   } }  package com.company;  public class Customer {  //Customer: id, Фамилия, Имя, Отчество, Адрес, Номер кредитной карточки, Номер банковского счета.  int id,num\_card,num\_sch,surname,name,patronymic,adr;   public Customer(int id) {  setId(id);  setNum\_card(id);  setNum\_sch(id);  setSurname(id);  setName(id);  setPatronymic(id);  setAdr(id);  }   @Override  public String toString() {  return "Customer{" +  "id=" + id +  ", surname=" + surname +  ", name=" + name +  ", patronymic=" + patronymic +  ", adr=" + adr +  ", num\_card=" + num\_card +  ", num\_sch=" + num\_sch +  '}';  }  public int getId() {  return id;  }  public int getNum\_card() {  return num\_card;  }  public int getNum\_sch() {  return num\_sch;  }  public int getSurname() {  return surname;  }  public int getName() {  return name;  }  public int getPatronymic() {  return patronymic;  }  public int getAdr() {  return adr;  }  public void setId(int id) {  this.id = id;  }  public void setNum\_card(int num\_card) {  this.num\_card = num\_card;  }  public void setNum\_sch(int num\_sch) {  this.num\_sch = num\_sch;  }  public void setSurname(int surname) {  this.surname = surname;  }  public void setName(int name) {  this.name = name;  }  public void setPatronymic(int patronymic) {  this.patronymic = patronymic;  }  public void setAdr(int adr) {  this.adr = adr;  } }  package com.company;  public class MyExc extends Throwable {  public void MyExc() {  System.out.println("Возникло исключение!!!");  } }  package com.company;  import java.util.Scanner;  public class Student { //Student: id, Фамилия, Имя, Отчество, Дата рождения, Адрес, Телефон, Факультет, Курс, Группа.  int id;  String surname,name,patronymic,date,adr,tel,faculty,course,group;   public Student(int id) {  Scanner in = new Scanner(System.in);  setId(id);  System.out.print("Введите фамилию студента - ");  setSurname(in.nextLine());  System.out.print("Введите имя студента - ");  setName(in.nextLine());  System.out.print("Введите отчество студента - ");  setPatronymic(in.nextLine());  System.out.print("Введите дату рождения студента - ");  setDate(in.nextLine());  System.out.print("Введите адрес студента - ");  setAdr(in.nextLine());  System.out.print("Введите телефон студента - ");  setTel(in.nextLine());  System.out.print("Введите факультет студента - ");  setFaculty(in.nextLine());  System.out.print("Введите курс студента - ");  setCourse(in.nextLine());  System.out.print("Введите группу студента - ");  setGroup(in.nextLine());  }  @Override  public String toString() {  return "Student{" +  "id=" + id +  ", surname='" + surname + '\'' +  ", name='" + name + '\'' +  ", patronymic='" + patronymic + '\'' +  ", date='" + date + '\'' +  ", adr='" + adr + '\'' +  ", tel='" + tel + '\'' +  ", faculty='" + faculty + '\'' +  ", course='" + course + '\'' +  ", group='" + group + '\'' +  '}';  }  public int getId() {  return id;  }  public String getSurname() {  return surname;  }  public String getName() {  return name;  }  public String getPatronymic() {  return patronymic;  }  public String getDate() {  return date;  }  public String getAdr() {  return adr;  }  public String getTel() {  return tel;  }  public String getFaculty() {  return faculty;  }  public String getCourse() {  return course;  }  public String getGroup() {  return group;  }  public void setId(int id) {  this.id = id;  }  public void setSurname(String surname) {  this.surname = surname;  }  public void setName(String name) {  this.name = name;  }  public void setPatronymic(String patronymic) {  this.patronymic = patronymic;  }  public void setDate(String date) {  this.date = date;  }  public void setAdr(String adr) {  this.adr = adr;  }  public void setTel(String tel) {  this.tel = tel;  }  public void setFaculty(String faculty) {  this.faculty = faculty;  }  public void setCourse(String course) {  this.course = course;  }  public void setGroup(String group) {  this.group = group;  } } |

Вариант 3:

1. В каждой строке найти и удалить заданную подстроку.
2. В каждой строке стихотворения Александра Блока найти и заменить заданную подстроку на подстроку иной длины.

Решение:

|  |
| --- |
| package com.company; import java.io.\*; import java.util.ArrayList; import java.util.Scanner;  public class Main {    public static ArrayList<String> choose\_enter(int choose){  ArrayList<String> list = new ArrayList<>();  switch (choose){  case 0:  Scanner in = new Scanner(System.in);  System.out.println("Введите имя начального файла (пункт 1)");  list.add(in.nextLine());  System.out.println("Введите абсолютный путь к начальному файлу (пункт 1)");  list.add(in.nextLine());  System.out.println("Введите имя начального файла (пункт 2)");  list.add(in.nextLine());  System.out.println("Введите абсолютный путь к начальному файлу (пункт 2)");  list.add(in.nextLine());  System.out.println("Введите имя нового файла (пункт 1)");  list.add(in.nextLine());  System.out.println("Введите абсолютный путь к новому файлу (пункт 1)");  list.add(in.nextLine());  System.out.println("Введите имя нового файла (пункт 2)");  list.add(in.nextLine());  System.out.println("Введите абсолютный путь к новому файлу (пункт 2)");  list.add(in.nextLine());  break;  case 1:  try {  File filepath = new File("/ooooooooooyyyyyyyyyyyyyyy/1.2/JAVA/Lab5/3/src/com/company/Pathname.txt");  FileReader fr = new FileReader(filepath);  Scanner scannerfile = new Scanner(fr);  while(scannerfile.hasNextLine()) {  list.add(scannerfile.nextLine());  }  fr.close();  } catch (FileNotFoundException e) {  e.printStackTrace();  } catch (IOException e) {  e.printStackTrace();  }  break;  default:  System.out.println("Неправильный ввод параметров!");  break;   }  return list;  }   public static ArrayList<String> read\_file (String pathname){  ArrayList<String> list = new ArrayList<>();  try {  File readfile = new File(pathname);  FileReader fr = new FileReader(readfile);  Scanner scannerfile = new Scanner(fr);  while(scannerfile.hasNextLine()) {  list.add(scannerfile.nextLine());  }  fr.close();  } catch (FileNotFoundException e) {  e.printStackTrace();  } catch (IOException e) {  e.printStackTrace();  }  return list;  }   public static void write\_file (ArrayList<String> list,String pathname){  File writefile = new File(pathname);  try {  FileWriter fr = new FileWriter(writefile, false);  for (int i = 0; i<list.size(); i++){  fr.write(list.get(i)+"\n");  }  fr.flush();  fr.close();  } catch (IOException e) {  e.printStackTrace();  }   System.out.println("Файл дополнен.");   }   public static void main(String[] args) {  /\*  В следующих заданиях требуется ввести последовательность строк из текстового потока и выполнить указанные действия.  При этом могут рассматриваться два варианта:  • каждая строка состоит из одного слова;  • каждая строка состоит из нескольких слов.  Имена входного и выходного файлов, а также абсолютный путь к ним могут быть введены как параметры командной строки или храниться в файле.  1. В каждой строке найти и удалить заданную подстроку.  2. В каждой строке стихотворения Александра Блока найти и заменить заданную подстроку на подстроку иной длины.  \*/  ArrayList<String> data,string\_of\_file,string\_of\_file2;  ArrayList<String> for\_new\_file = new ArrayList<>();  ArrayList<String> for\_new\_file2 = new ArrayList<>();  Scanner in = new Scanner(System.in);  System.out.println("Выберете способ введения имен входного и выходного файлов, а также абсолютный путь к ним: 0 - введение вручную; 1 - из файла");  int choose = in.nextInt();  data = choose\_enter(choose);  System.out.println("------------------------");  System.out.println("№ 1");  string\_of\_file = read\_file(data.get(1));  System.out.println(string\_of\_file);  for (int i = 0; i<string\_of\_file.size(); i++){  for\_new\_file.add(string\_of\_file.get(i).replace("один",""));  }  write\_file(for\_new\_file,data.get(5));  System.out.println("------------------------");  System.out.println("№ 2");  string\_of\_file2 = read\_file(data.get(3));  for (int i = 0; i<string\_of\_file2.size(); i++){  for\_new\_file2.add(string\_of\_file2.get(i).replace("фонарь","фонарик123456789"));  }  write\_file(for\_new\_file2,data.get(7));  } } |

Вариант 4:

1. Прочитать текст Java-программы и все слова public в объявлении атрибутов и методов класса заменить на слово private.
2. Прочитать текст Java-программы и записать в другой файл в обратном порядке символы каждой строки.

Решение:

|  |
| --- |
| package com.company; import java.io.IOException; import java.io.\*; import java.util.ArrayList; import java.util.Scanner;  public class Main {   public static ArrayList<String> read\_file (){  String pathname = "/ooooooooooyyyyyyyyyyyyyyy/1.2/JAVA/Lab5/4/src/com/company/Java.txt";  ArrayList<String> list = new ArrayList<>();   try {  File readfile = new File(pathname);  if (readfile.exists()){  FileReader fr = new FileReader(readfile);  Scanner scannerfile = new Scanner(fr);  while(scannerfile.hasNextLine()) {  list.add(scannerfile.nextLine());  }  fr.close();  }else{  System.out.println("Not file");  }   } catch (FileNotFoundException e) {  e.printStackTrace();  } catch (IOException e) {  e.printStackTrace();  }  return list;  }   public static void write\_file (ArrayList<String> list,String pathname){  File writefile = new File(pathname);  if (writefile.exists()){  try {  FileWriter fr = new FileWriter(writefile, false);  for (int i = 0; i<list.size(); i++){  fr.write(list.get(i)+"\n");  }  fr.flush();  fr.close();  } catch (IOException e) {  e.printStackTrace();  }   System.out.println("Файл дополнен.");  }  else {  System.out.println("Problem");  }  }   public static void main(String[] args) {  /\*  Вариант 4 При выполнении следующих заданий для вывода результатов создавать новую директорию и файл средствами класса File 1. Прочитать текст Java-программы и все слова public в объявлении атрибутов и методов класса заменить на слово private. 2. Прочитать текст Java-программы и записать в другой файл в обратном порядке символы каждой строки.  \*/  ArrayList<String> file;  file = read\_file();  System.out.println("------------------------");  System.out.println("№ 1");  ArrayList<String> for\_new\_file1 = new ArrayList<>();  for (int i = 0; i<file.size(); i++){  for\_new\_file1.add(file.get(i).replace("public","private"));  }  write\_file(for\_new\_file1,"/ooooooooooyyyyyyyyyyyyyyy/1.2/JAVA/Lab5/4/src/com/company/New\_file1.txt");  System.out.println("------------------------");  System.out.println("№ 2");  ArrayList<String> for\_new\_file2 = new ArrayList<>();  for (int i = 0; i<file.size(); i++){  String obr\_str = new StringBuffer(file.get(i)).reverse().toString();  for\_new\_file2.add(obr\_str);  }  write\_file(for\_new\_file2,"/ooooooooooyyyyyyyyyyyyyyy/1.2/JAVA/Lab5/4/src/com/company/New\_file2.txt");  } } |

Ссылка на репозиторий:

https://github.com/Carpediem2026/BigData