|  |  |
| --- | --- |
| 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 № 3** |

**Название:**

Классы

**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:

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 методы для вычисления модуля вектора, скалярного произведения, сложения, вычитания, умножения на константу. Объявить массив объектов. Написать метод, который для заданной пары векторов будет определять, являются ли они коллинеарными или ортогональными.

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("Вариант 1");  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();  }  } }  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:

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.

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();  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());}  }    System.out.println("------------------------------");  System.out.println("Вариант 2.2");  System.out.print("Введите количество Customer - ");  int n2 = in.nextInt();  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());}  }   } }  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;  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. Создать объект класса Текстовый файл, используя класс Файл. Методы: создать, переименовать, вывести на консоль содержимое, дополнить, удалить.
2. Создать объект класса Одномерный массив, используя класс Массив. Методы: создать, вывести на консоль, выполнить операции (сложить, вычесть, перемножить).

Решение:

|  |
| --- |
| package com.company; import java.io.IOException; import com.company.TextFile; import com.company.OneArray;   public class Main {   public static void main(String[] args) {  /\*  Вариант 3  Создать приложение, удовлетворяющее требованиям, приведенным в задании.  Аргументировать принадлежность классу каждого создаваемого метода и корректно переопределить для каждого класса методы equals(), hashCode(), toString().  1. Создать объект класса Текстовый файл, используя класс Файл. Методы: создать, переименовать, вывести на консоль содержимое, дополнить, удалить.  2. Создать объект класса Одномерный массив, используя класс Массив. Методы: создать, вывести на консоль, выполнить операции (сложить, вычесть, перемножить).  \*/  //Scanner in = new Scanner(System.in);  System.out.println("------------------------------");  System.out.println("Вариант 3.1");  TextFile file = new TextFile("/Users/63462/Downloads/files.txt");  System.out.println(file);  file.rename();  System.out.println(file);  try {  file.write();  file.print();  } catch (IOException e){  System.out.println("no print/ no write");  }  System.out.println(" ");  file.delFile();  System.out.println("------------------------------");  System.out.println("Вариант 3.2");  OneArray myarr = new OneArray(10);   } }  package com.company; //import java.util.Array;  import java.util.ArrayList; import java.util.Objects;  public class OneArray{  int lenghtArray;  //int[] array = new int[lenghtArray];  ArrayList<Integer> summ = new ArrayList<Integer>();  ArrayList<Integer> del = new ArrayList<Integer>();  ArrayList<Integer> mno = new ArrayList<Integer>();   void create (){  int[] array = new int[lenghtArray];  for (int i = 0; i < array.length; i++) {  array[i] = ((int)(Math.random() \* 50) - 20);  //System.out.print(array[i]+" ");  this.print(array[i]);  this.algbr(i,array[i]);  }  System.out.println(" ");  System.out.println("Сумма - "+summ);  System.out.println("Разница - "+del);  System.out.println("произведение - "+mno);   }  public void print (int a){  System.out.print(a+" ");  }  public void algbr(int i,int a){  summ.add(i,(a+10));  del.add(i,(a-100));  mno.add(i,(a\*5));  }   public OneArray(int lenghtArray) {  this.lenghtArray = lenghtArray;  this.create();  }   @Override  public boolean equals(Object o) {  if (this == o) return true;  if (!(o instanceof OneArray)) return false;  OneArray oneArray = (OneArray) o;  return lenghtArray == oneArray.lenghtArray && Objects.equals(summ, oneArray.summ) && Objects.equals(del, oneArray.del) && Objects.equals(mno, oneArray.mno);  }   @Override  public int hashCode() {  return Objects.hash(lenghtArray);  } }  package com.company; import java.io.File; import java.io.\*; import java.nio.file.Files; import java.nio.file.Paths; import java.util.Objects;  public class TextFile extends File {  String pathname;   //Методы: создать, переименовать, вывести на консоль содержимое, дополнить, удалить.  void create(){  String data = "Это тестовая строка для записи в файл";  try {  Files.write(Paths.get(pathname), data.getBytes());  } catch (IOException e) {  e.printStackTrace();  }  }  void rename(){  File renameFile = new File("/Users/63462/Downloads/renamefiles.txt");  if(this.renameTo(renameFile)){  System.out.println("Файл переименован успешно");;  }else{  System.out.println("Файл не был переименован");  }  //this.pathname = "/Users/63462/Downloads/renamefiles.txt";  }  void print () throws IOException {  FileReader reader = new FileReader(this);  int c;  while ((c= reader.read())!=-1){  System.out.print((char)c);  }  reader.close();  }  void write() throws IOException{  FileWriter writer = new FileWriter(this,true);  String dopText = ". Дополнение файла прошло успешно";  writer.write(dopText);  writer.flush();  writer.close();  System.out.println("Файл дополнен.");  }  void delFile(){  if(this.delete()){  System.out.println("Файл удален.");  }else System.out.println("Файла не удален.");  }    public TextFile(String pathname) {  super(pathname);  this.pathname = pathname;  create();  }   @Override  public String toString() {  return "TextFile{" +  "pathname='" + pathname +  '}';  }   @Override  public boolean equals(Object o) {  if (this == o) return true;  if (!(o instanceof TextFile)) return false;  if (!super.equals(o)) return false;  TextFile textFile = (TextFile) o;  return Objects.equals(pathname, textFile.pathname);  }   @Override  public int hashCode() {  return Objects.hash(super.hashCode(), pathname);  } } |

Вариант 4:

1. Система Факультатив. Преподаватель объявляет запись на Курс. Студент записывается на Курс, обучается и по окончании Преподаватель выставляет Оценку, которая сохраняется в Архиве. Студентов, Преподавателей и Курсов при обучении может быть несколько.
2. Система Платежи. Клиент имеет Счет в банке и Кредитную Карту (КК). Клиент может оплатить Заказ, сделать платеж на другой Счет, заблокировать КК и аннулировать Счет. Администратор может заблокировать КК за превышение кредита.

Решение:

|  |
| --- |
| package com.company;  public class Main {   public static void main(String[] args) {  /\*  Построить модель программной системы.  1. Система Факультатив. Преподаватель объявляет запись на Курс.  Студент записывается на Курс, обучается и по окончании Преподаватель выставляет Оценку,  которая сохраняется в Архиве. Студентов, Преподавателей и Курсов при обучении может быть несколько.  2. Система Платежи. Клиент имеет Счет в банке и Кредитную Карту (КК).  Клиент может оплатить Заказ, сделать платеж на другой Счет, заблокировать КК и аннулировать Счет.  Администратор может заблокировать КК за превышение кредита.  \*/  System.out.println("------------------------------");  System.out.println("Вариант 4.1");  int m =2;  Cours[] cours = new Cours[m];  for (int i = 0; i < m; i++) {  cours[i] = new Cours(i);  cours[i].createfile();  }   System.out.println("------------------------------");  System.out.println("Вариант 4.2");  Client client = new Client("Client 123",123);  client.status();  client.pay(1000);  client.status();  client.payto(100);  client.status();  client.blockKKandNULL();  client.status();  } }  package com.company;  public class Account {  int id,summ;   public Account(int id, int summ) {  this.id = id;  this.summ = summ;  }  public int getId() {  return id;  }   public int getSumm() {  return summ;  }   public void setSumm(int summ) {  this.summ = summ;  }   @Override  public String toString() {  return "Account{" +  "id=" + id +  ", summ=" + summ +  '}';  } }  package com.company;  public class Client {  Account acc = new Account(1,3000);  KK card = new KK(1,1000);  String fio;  int id;   public Client(String fio, int id) {  this.fio = fio;  this.id = id;  }  void pay(int s){  acc.setSumm(acc.getSumm()-s);  System.out.println("Pay on 1000 (account): ost\_acc - "+acc.getSumm());  }  void payto(int s){  if (card.getBlock()){  System.out.println("Card in block");  }else{  acc.setSumm(acc.getSumm()-s);  card.setSumm(card.getSumm()+s);  System.out.println("Pay acc to kk 100: ost\_acc - "+acc.getSumm()+"; ost\_kk - "+card.getSumm());  }  }  void blockKKandNULL(){  card.setBlock(true);  this.payto(100);  acc.setSumm(0);  }  void status(){  System.out.println(card);  System.out.println(acc);  } }  package com.company; import java.io.File; import java.io.FileWriter; import java.io.IOException; import java.nio.file.Files; import java.nio.file.Paths; import java.util.Arrays;  public class Cours {  int idCours;  Teach[] teaches = new Teach[2];  public Cours(int idCours) {  this.idCours = idCours;  for (int i = 0; i < 2; i++) {  teaches[i] = new Teach(i,idCours);  }  }  public void createfile (){  String pathname = "/Users/63462/Downloads/arxivForCours"+idCours+".txt";  File file = new File(pathname);  String data = this.toString();  try {  Files.write(Paths.get(pathname), data.getBytes());  } catch (IOException e) {  e.printStackTrace();  }  }   @Override  public String toString() {  return "Cours{" +  "idCours=" + idCours +  ", teaches=" + Arrays.toString(teaches) +  '}';  } }  package com.company;  public class KK {  int id,summ;  boolean block;  public KK(int id, int summ) {  this.id = id;  this.summ = summ;  this.block = false;  }   public int getId() {  return id;  }   public int getSumm() {  return summ;  }   public void setBlock(boolean block) {  this.block = block;  }   public boolean getBlock() {  return block;  }   public void setSumm(int summ) {  this.summ = summ;  }   @Override  public String toString() {  return "KK{" +  "id=" + id +  ", summ=" + summ +  ", status\_block=" + block +  '}';  } }  package com.company;  public class Student {  int idStudent,idTeach, idCours, score;  String FIO;  public Student(int idCours, int idTeach, int idStudent) {  this.idStudent = idStudent;  this.idTeach = idCours;  this.idTeach = idTeach;  this.FIO = idTeach+" "+idStudent+" "+idStudent;  this.score = (int)(Math.random()\*6);  }   @Override  public String toString() {  return "Student{" +  "idStudent=" + idStudent +  ", idTeach=" + idTeach +  ", idCours=" + idCours +  ", score=" + score +  ", FIO='" + FIO + '\'' +  '}';  } }  package com.company; import java.util.Arrays;  public class Teach {  int idTeach;  int idCours;  String FIO;  Student[] students = new Student[2];  public Teach(int idTeach, int idCours) {  this.idTeach = idTeach;  this.idCours = idCours;  this.FIO = idTeach+" "+idTeach+" "+idTeach;  for (int i = 0; i < 2; i++) {  students[i] = new Student(idCours,idTeach,i);  }  }  @Override  public String toString() {  return "Teach{" +  "idTeach=" + idTeach +  ", idCours=" + idCours +  ", FIO='" + FIO + '\'' +  ", students=" + Arrays.toString(students) +  '}';  } } |

Ссылка на репозиторий:

https://github.com/Carpediem2026/BigData