МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ   
ВЫСШЕГО ОБРАЗОВАНИЯ «САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЁВА (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)» (САМАРСКИЙ УНИВЕРСИТЕТ)**

Институт информатики и кибернетики  
Кафедра программных систем

Дисциплина  
**Спецкурс по языку Ява**

**ОТЧЕТ**по лабораторной работе № 2 **«Синтаксис языка»**

Студент: Кочетков Е.В.  
Группа: 6202-020302D  
  
Преподаватель: Ледкова Т.А  
  
Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Самара 2024

**Задание**

-------Program.java-------

import java.util.Scanner;  
  
public class Program {  
  
 public static void main(String[] s) {  
  
  
 System.***out***.println("=====СОЗДАНИЕ ВЕКТОРА=====");  
  
 try {  
 System.***out***.print("Введите размерность вектора: ");  
 Scanner scan = new Scanner(System.***in***);  
 int n = Integer.*parseInt*(scan.nextLine());  
 Vector vec = new Vector(n);  
 for (int i = 0; i < vec.getLen(); i++) {  
 System.***out***.print("Введите значение [" + i + "]: ");  
 double k = Double.*parseDouble*(scan.nextLine());  
 vec.setCord(i, k);  
 }  
  
 System.***out***.println("=====МАКСИМАЛЬНОЕ И МИНИМАЛЬНОЕ ЗНАЧЕНИЯ=====");  
  
 System.***out***.println("Максимальное значение: " + vec.maxCord());  
 System.***out***.println("Минимальное значение: " + vec.minCord());  
  
 System.***out***.println("=====СОРТИРОВКА МАССИВА=====");  
 System.***out***.println("Массив до сортировки");  
 vec.printVector();  
  
 Vector sorted = new Vector(vec.getLen());  
 for (int i = 0; i < vec.getLen(); i++) {  
 sorted.setCord(i, vec.getCord(i));  
 }  
 sorted.sortVector();  
 System.***out***.println("Массив после сортировки");  
 sorted.printVector();  
  
 sorted = null;  
  
 System.***out***.println("=====УМНОЖЕНИЕ НА ЧИСЛО=====");  
 System.***out***.print("Введите число: ");  
 double k = Double.*parseDouble*(scan.nextLine());  
 System.***out***.println("Вектор до умножения:");  
 vec.printVector();  
 System.***out***.println("Вектор после умножения:");  
 (vec.mulNumber(k)).printVector();  
  
 System.***out***.println("=====ЕВКЛИДОВО ЧИСЛО=====");  
  
 System.***out***.print("Введите размерность вектора 2: ");  
 n = Integer.*parseInt*(scan.nextLine());  
 Vector vec2 = new Vector(n);  
 for (int i = 0; i < vec.getLen(); i++) {  
 System.***out***.print("Введите значение [" + i + "]: ");  
 k = Double.*parseDouble*(scan.nextLine());  
 vec2.setCord(i, k);  
 }  
 if (vec.getLen() == vec2.getLen()) {  
 System.***out***.println("Евклидово число: " + vec.evklid(vec2));  
  
 System.***out***.println("=====СУММА ВЕКТОРОВ=====");  
 System.***out***.println("Вектор до сложение:");  
 vec.printVector();  
 System.***out***.println("Вектор после сложения:");  
 (Vector.*sumVector*(vec, vec2)).printVector();  
  
 System.***out***.println("=====СКАЛЯРНОЕ ПРОИЗВЕДЕНИЕ=====");  
 System.***out***.println("Скалярное произведение: " + Vector.*scalar*(vec, vec2));  
  
 } else {  
 System.***out***.println("Ошибка! Размерности векторов не совпадают!");  
 }  
 } catch (RuntimeException e) {  
 System.***out***.println("Ошибка! " + e.getMessage());  
 }  
 }  
}

-------Vector.java-------  
public class Vector {  
  
 private final double[] **vector**;  
  
 public Vector(int n) { // конструктор, что задает длину массива  
  
 **vector** = new double[n];  
 }  
  
 public double getCord(int n) { // получение элемента массива  
 if (n < 0 || n >= **vector**.**length**) {  
 throw new IndexOutOfBoundsException("Ошибка! Такой координаты у вектора нет!"); // ***TODO: 15.02.2024 что-то с этим сделать*** } else {  
 return **vector**[n];  
 }  
 }  
  
 public void setCord(int n, double i) { // задание элемента массива  
 if (n < 0 || n > **vector**.**length**) {  
 System.***out***.println("Ошибка! Такой координаты у вектора нет!");  
 } else {  
 **vector**[n] = i;  
 }  
 }  
  
 public int getLen() { // длина массива  
 return **vector**.**length**;  
 }  
  
 public double maxCord() { // максимальное число в массиве  
 double maxCord = **vector**[0];  
 for (int i = 0; i < **vector**.**length**; i++)  
 if (**vector**[i] > maxCord) {  
 maxCord = **vector**[i];  
 }  
 return maxCord;  
 }  
  
 public double minCord() { // минимальное число в массиве  
 double minCord = **vector**[0];  
 for (int i = 0; i < **vector**.**length**; i++) {  
 if (**vector**[i] < minCord) {  
 minCord = **vector**[i];  
 }  
 }  
 return minCord;  
 }  
  
 public void sortVector() {  
  
  
  
 for (int i = 0; i < **vector**.**length**; i++) {  
 for (int j = i; j < **vector**.**length**; j++) {  
 if (**vector**[i] > **vector**[j]) {  
 double temp = **vector**[j];  
 **vector**[j] = **vector**[i];  
 **vector**[i] = temp;  
 }  
 }  
 }  
 }  
  
 public double evklid(Vector vec2) {  
  
 if (**vector**.**length** != vec2.getLen()) { // проверка, чтобы размерности совпадали  
 System.***out***.println("Ошибка! Размерности векторов не совпадают!");  
 return 0;  
 } else {  
 double res = 0;  
 for (int i = 0; i < **vector**.**length**; i++) {  
 res += Math.*pow*(**vector**[i]-vec2.getCord(i), 2);  
 }  
 res = Math.*sqrt*(res);  
 return res;  
 }  
 }  
  
 public Vector mulNumber(double num) { // умножение на число  
 Vector res = new Vector(**vector**.**length**);  
 for (int i = 0; i < **vector**.**length**; i++) {  
 res.setCord(i, **vector**[i] \* num);  
 }  
 return res;  
 }  
  
 public static Vector sumVector(Vector vec1, Vector vec2) { // сумма векторов  
 if (vec1.getLen() != vec2.getLen()) { // проверка, чтобы размерности совпадали  
 System.***out***.println("Ошибка! Размерности векторов не совпадают!");  
 return null;  
 } else {  
 Vector res = new Vector(vec1.getLen());  
 for (int i = 0; i < vec1.getLen(); i++) {  
 res.setCord(i, vec1.getCord(i) + vec2.getCord(i));  
 }  
 return res;  
 }  
 }  
   
  
 public static double scalar(Vector vec1, Vector vec2) // скалярное произведение  
 {  
 if (vec1.getLen() != vec2.getLen()) { // проверка, чтобы размерности совпадали  
 System.***out***.println("Ошибка! Размерности векторов не совпадают!");  
 return 0;  
 } else {  
 double res = 0;  
 for (int i = 0; i < vec1.getLen(); i++) {  
 res += vec1.getCord(i) \* vec2.getCord(i);  
 }  
 return res;  
 }  
 }  
  
  
  
 public void printVector() {  
 for (int i = 0; i < **vector**.**length**; i++) {  
 System.***out***.print(**vector**[i] + " ");  
 }  
 System.***out***.println();  
 }  
  
}



