



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 \_\_\_\_\_ «Теоретическая информатика и компьютерные технологии»

**Лабораторная работа № 2**  
**по курсу «Языки и методы программирования»**  
**«Разработка простейшего класса на языке Java»**

Студент группы ИУ9-21Б Горбунов А. Д.

Преподаватель Посевин Д. П.

*Москва 2023*

# 1 Задание

Выполнение лабораторной работы заключается в составлении на языке Java одного из классов, приведённых в таблице. В классе обязательно должны присутствовать конструктор и метод toString.

Отладку разработанного класса необходимо осуществить в методе main вспомогательного класса Test. Использование контейнерных классов из стандартной библиотеки языка Java не разрешается.

Класс n-мерных вещественных векторов с операциями сложения, вычитания и нормализации. Любая операция должна возвращать новый вектор, оставляя исходные вектора нетронутыми.

## 2 Результаты

Исходный код программы представлен в листинге 1, 2, 3.

Листинг 1 — класс Test

```
1 import java.util.Scanner;
2 public class Test {
3     static int n = 4;
4     public static void main(String[] args) {
5         double[] arr_1 = new double[n];
6         arr_1[0] = 5; arr_1[1] = 3; arr_1[2] = 1;
7         Vector vec_1 = new Vector(n, arr_1);
8         Vector vec_2 = new Vector(n, arr_1);
9         System.out.println("vec_1:");
10        vec_1.printVector();
11        System.out.println("vec_2:");
12        vec_2.printVector();
13        Vector vec_new_1 = vec_1.sum(vec_2);
14        System.out.println("vec_new_1 = vec_1.sum(vec_2):");
15        vec_new_1.printVector();
16        System.out.println("vec_1:");
17        vec_1.printVector();
18        System.out.println("vec_2:");
19        vec_2.printVector();
20
21        Vector vec_new_2 = vec_1.subtraction(vec_2);
22        System.out.println("vec_new_2 = vec_1.subtraction(vec_2):");
23        vec_new_2.printVector();
24        System.out.println("vec_1:");
25        vec_1.printVector();
26        System.out.println("vec_2:");
27        vec_2.printVector();
```

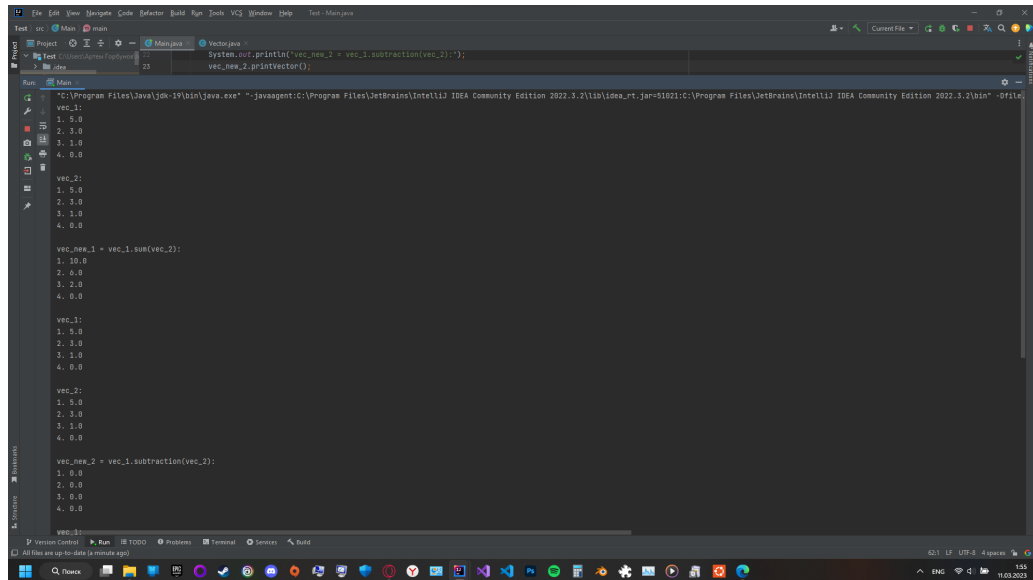
## Листинг 2 — класс Test(продолжение)

```
1      Vector vec_new_3 = vec_new_1.normalization();
2      System.out.println("vec_new_3 = vec_new_1.normalization() :");
3      vec_new_3.printVector();
4      System.out.println("vec_new_1:");
5      vec_new_1.printVector();
6      //////////////////////////////////////
7      Scanner scan = new Scanner(System.in);
8      double[] arr_2 = new double[n];
9      double[] arr_3 = new double[n];
10
11     for(int i = 0; i < n; i++)
12     {
13         if(scan.hasNextDouble())
14             arr_2[i] = scan.nextDouble();
15         else
16         {
17             System.out.println(" Error");
18             break;
19         }
20     }
21     Vector vec_3 = new Vector(n, arr_2);
22
23     for(int i = 0; i < n; i++)
24     {
25         if(scan.hasNextDouble())
26             arr_3[i] = scan.nextDouble();
27         else
28         {
29             System.out.println(" Error");
30             break;
31         }
32     }
33     Vector vec_4 = new Vector(n, arr_3);
34     System.out.println("vec_3:");
35     vec_3.printVector();
36     System.out.println("vec_4:");
37     vec_4.printVector();
38     Vector vec_sum_vec_1__vec_3 = vec_3.sum(vec_4);
39     System.out.println("vec_sum_vec_1__vec_3:");
40     vec_sum_vec_1__vec_3.printVector();
41
42 }
43 }
```

### Листинг 3 — класс Vector

```
1 import static java.lang.Math.*;
2 public class Vector {
3     static private int n;
4     private double[] coordinates;
5     public Vector(int n, double[] coordinates)
6     {
7         this.n = n;
8         this.coordinates = new double[this.n];
9         for(int i = 0; i < n; i++)
10             this.coordinates[i] = 0;
11         for(int i = 0; i < coordinates.length; i++)
12             this.coordinates[i] = coordinates[i];
13     }
14     public Vector sum(Vector vec_2)
15     {
16         double[] sup_coordinates = new double[this.n];
17         for(int i = 0; i < n; i++)
18             sup_coordinates[i] = coordinates[i] + vec_2.coordinates[i];
19         Vector new_vec = new Vector(sup_coordinates.length,
20 sup_coordinates);
21         return new_vec;
22     }
23     public Vector subtraction(Vector vec_2)
24     {
25         double[] sup_coordinates = new double[this.n];
26         for(int i = 0; i < n; i++)
27             sup_coordinates[i] = coordinates[i] - vec_2.coordinates[i];
28         Vector new_vec = new Vector(sup_coordinates.length,
29 sup_coordinates);
30         return new_vec;
31     }
32     public Vector normalization()
33     {
34         double[] sup_coordinates = new double[this.n];
35         double size = size();
36         for(int i = 0; i < n; i++)
37             sup_coordinates[i] = coordinates[i]/size;
38         Vector new_vec = new Vector(sup_coordinates.length,
39 sup_coordinates);
40         return new_vec;
41     }
42     public double size()
43     {
44         double sum = 0;
45         for(int i = 0; i < n; i++)
46             sum += this.coordinates[i] * this.coordinates[i];
47         return sqrt(sum);
48     }
49     public void printVector()
50     {
51         for(int i = 0; i < n; i++)
52             System.out.println(i + 1 + ". " + coordinates[i]);
53         System.out.println();
54     }
55 }
```

Результат запуска представлен на рисунке 1, 2, 3.



```
Test - Main.java
Run
vec_1:
1. 5.0
2. 3.0
3. 1.0
4. 0.0

vec_2:
1. 5.0
2. 3.0
3. 1.0
4. 0.0

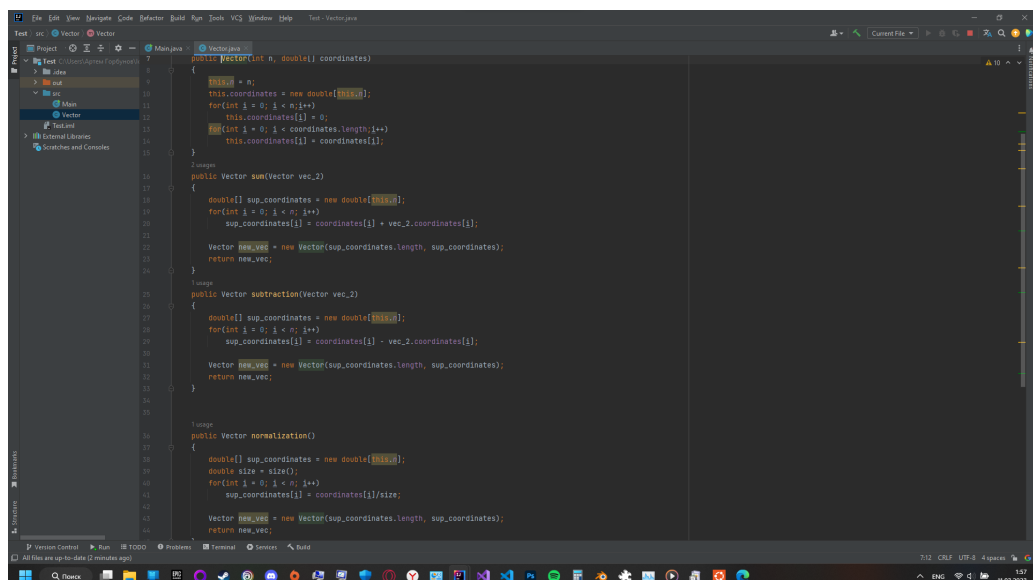
vec_new_1 = vec_1.sub(vec_2):
1. 0.0
2. 0.0
3. 2.0
4. 0.0

vec_1:
1. 5.0
2. 3.0
3. 1.0
4. 0.0

vec_2:
1. 5.0
2. 3.0
3. 1.0
4. 0.0

vec_new_2 = vec_1.subtraction(vec_2):
1. 0.0
2. 0.0
3. 0.0
4. 0.0
```

Рис. 1 — Вывод программы



```
Test - Vector.java
Vector.java
public Vector(int n, double[] coordinates) {
    this.n = n;
    this.coordinates = new double[n];
    for (int i = 0; i < n; i++) {
        this.coordinates[i] = 0;
    }
    for (int i = 0; i < coordinates.length; i++) {
        this.coordinates[i] = coordinates[i];
    }
}

2 usages
public Vector sum(Vector vec_2) {
    double[] sup_coordinates = new double[this.n];
    for (int i = 0; i < n; i++) {
        sup_coordinates[i] = coordinates[i] + vec_2.coordinates[i];
    }
    Vector new_vec = new Vector(sup_coordinates.length, sup_coordinates);
    return new_vec;
}

1 usage
public Vector subtraction(Vector vec_2) {
    double[] sup_coordinates = new double[this.n];
    for (int i = 0; i < n; i++) {
        sup_coordinates[i] = coordinates[i] - vec_2.coordinates[i];
    }
    Vector new_vec = new Vector(sup_coordinates.length, sup_coordinates);
    return new_vec;
}

1 usage
public Vector normalization() {
    double[] sup_coordinates = new double[this.n];
    double size = size();
    for (int i = 0; i < n; i++) {
        sup_coordinates[i] = coordinates[i]/size;
    }
    Vector new_vec = new Vector(sup_coordinates.length, sup_coordinates);
    return new_vec;
}
```

Рис. 2 — Работа программы

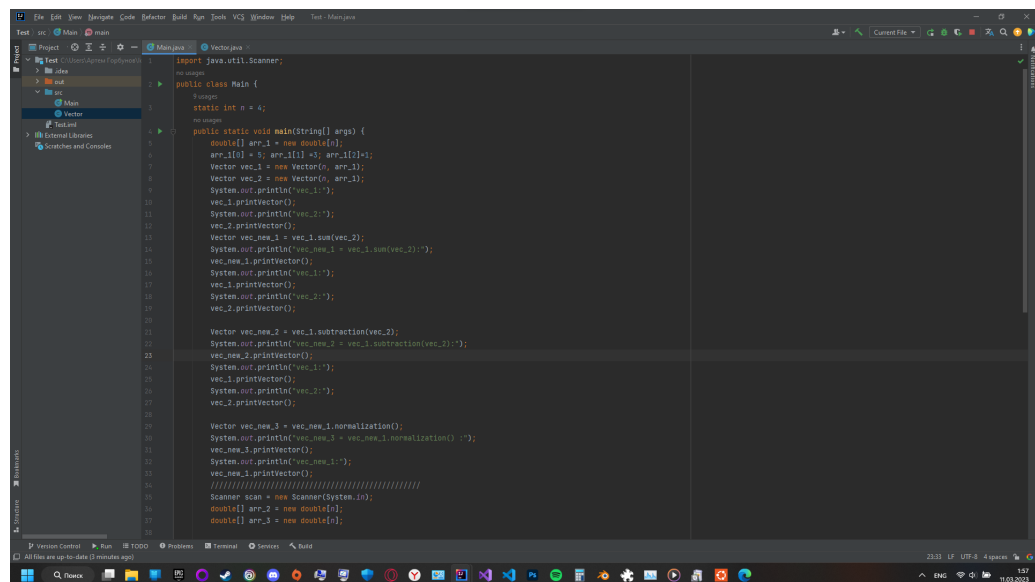


Рис. 3 — Работа программы