



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, искусственный интеллект и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Летучка № 2
по курсу «Языки и методы программирования»
«Модель вселенной»

Студент группы ИУ9-21Б Яннаев А. С.

Преподаватель Посевин Д. П.

Москва 2025

1 Цель работы

Реализовать модель вселенной.

2 Задание

Каждый элемент вселенной должен быть объектом некоего публичного класса, который инициализируется вспомогательным публичным классом порождающим эту вселенную. При инициализации экземпляров класса частиц моделируемой вселенной необходимо подсчитывать количество частиц вселенной используя статичное экземплярное поле защищенное от изменения из объектов внешних классов путем реализации статичного метода. Сформировать исходные данные и определить необходимые экземплярные поля для хранения состояния объектов частиц вселенной в соответствии с условием задачи и реализовать расчет. Программа должна обладать консольным пользовательским интерфейсом ввода данных, например, количества частиц вселенной, масс частиц и т.д.

Вариант №12 | Оценить расстояние между двумя вселенными

3 Результаты

Исходный код программы представлен в 1, 2, 3, 4.

Результат запуска представлен на рисунке 1.

Листинг 1 — Файл Main.java

```
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("particles in 1st: ");
8         int n1 = sc.nextInt();
9         Universe u1 = new Universe(n1);
10        for (int i = 0; i < n1; i++) {
11            System.out.print(" x, y, z for " + (i + 1) + ": ");
12            double x = sc.nextDouble();
13            double y = sc.nextDouble();
14            double z = sc.nextDouble();
15            u1.addParticle(i, x, y, z);
16        }
17
18        System.out.print("particles in 2nd: ");
19        int n2 = sc.nextInt();
20        Universe u2 = new Universe(n2);
21        for (int i = 0; i < n2; i++) {
22            System.out.print("x, y, z for " + (i + 1) + ": ");
23            double x = sc.nextDouble();
24            double y = sc.nextDouble();
25            double z = sc.nextDouble();
26            u2.addParticle(i, x, y, z);
27        }
28
29        double distance = Calculator.getDistance(u1, u2);
30        System.out.println("distance: " + distance);
31        System.out.println("particles: " + Particle.getCount());
32
33        sc.close();
34    }
35 }
```

ЛИСТИНГ 2 — Файл Particle.java

```
1 public class Particle {
2     private double x, y, z;
3     private static int count = 0;
4
5     public Particle(double x, double y, double z) {
6         this.x = x;
7         this.y = y;
8         this.z = z;
9         count++;
10    }
11
12    public static int getCount() {
13        return count;
14    }
15
16    public double[] getPosition() {
17        return new double[] {x, y, z};
18    }
19 }
```

ЛИСТИНГ 3 — Файл Universe.java

```
1 public class Universe {
2     private Particle[] particles;
3
4     public Universe(int size) {
5         particles = new Particle[size];
6     }
7
8     public void addParticle(int index, double x, double y, double z) {
9         particles[index] = new Particle(x, y, z);
10    }
11
12    public double[] getCenter() {
13        double sumX = 0, sumY = 0, sumZ = 0;
14        for (Particle p : particles) {
15            double[] pos = p.getPosition();
16            sumX += pos[0];
17            sumY += pos[1];
18            sumZ += pos[2];
19        }
20        int n = particles.length;
21        return new double[] {sumX / n, sumY / n, sumZ / n};
22    }
23 }
```

Листинг 4 — Файл Particle.java

```
1 public class Calculator {  
2     public static double getDistance(Universe u1, Universe u2) {  
3         double[] c1 = u1.getCenter();  
4         double[] c2 = u2.getCenter();  
5         return Math.sqrt(Math.pow(c1[0] - c2[0], 2) + Math.pow(c1[1] -  
6         c2[1], 2) + Math.pow(c1[2] - c2[2], 2));  
7     }  
}
```

```
/home/keiichi/jdk-23.0.2/bin/java -javaagent:/home/keiichi/Зарп  
введите количество частиц в первой вселенной: 3  
введите координаты x, y, z для частицы 1: 1 2 3  
введите координаты x, y, z для частицы 2: 2 3 4  
введите координаты x, y, z для частицы 3: 3 4 5  
введите количество частиц во второй вселенной: 3  
введите координаты x, y, z для частицы 1: 5 4 3  
введите координаты x, y, z для частицы 2: 4 3 2  
введите координаты x, y, z для частицы 3: 3 2 1  
расстояние между центрами вселенных: 2.8284271247461903  
общее количество частиц: 6  
  
Process finished with exit code 0
```

Рис. 1 — Результат