



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

Студент группы ИУ9-21Б Горбунов А. Д.

Преподаватель Посевин Д. П.

Москва 2023

1 Задание

Реализовать модель вселенной. 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.

Определить расстояние между двумя вселенными

2 Результаты

Исходный код программы представлен в листингах 1, 2, 3, 4, 5.

Листинг 1 — Нахождение расстояние между двумя вселенными(класс Test)

```
1 import java.util.Scanner;
2 public class Test {
3     private static final int n = 10;
4     public static void main(String[] args) {
5         Scanner scan = new Scanner(System.in);
6
7         Universe univ1 = new Universe(n);
8         univ1.newParticlesUniverse(1,1,1);
9         univ1.newParticlesUniverse(2,2,2);
10        System.out.println(univ1.getX() +" "+univ1.getY() +" "+univ1.
getZ());
11        Universe univ2 = new Universe(n);
12        univ2.newParticlesUniverse(0,0,0);
13        univ2.newParticlesUniverse(-1,-1,-1);
14        System.out.println(univ2.getX() +" "+univ2.getY() +" "+univ2.
getZ());
15        double result = univ1.distanceBetweenUniverses(univ2);
16        System.out.println(result);
17        ///////////////////////////////////
18        Universe univ3 = new Universe(n);
19        univ3.newParticlesUniverse(scan.nextInt(),scan.nextInt(),scan.
nextInt());
20        univ3.newParticlesUniverse(scan.nextInt(),scan.nextInt(),scan.
nextInt());
21        System.out.println(univ3.getX() +" "+univ3.getY() +" "+univ3.
getZ());
```

Листинг 2 — Нахождение расстояния между двумя вселенными (Класс Test)(продолжение)

```
1      Universe univ4 = new Universe(n);
2      univ4.newParticlesUniverse(scan.nextInt(), scan.nextInt(), scan.
nextInt());
3      univ4.newParticlesUniverse(scan.nextInt(), scan.nextInt(), scan.
nextInt());
4      System.out.println(univ4.getX() + " "+univ4.getY() + " "+univ4.
getZ());
5
6      System.out.println(result);
7  }
8 }
```

Результат запуска представлен на рисунке 1.

Листинг 3 — Нахождение расстояния между двумя вселенными (Класс Universe)

```
1 import static java.lang.Math.*;
2 public class Universe {
3     private static int number_universes;
4     private final int particles;
5     private int number_particles_this_universe = 0;
6     private double x, y, z;
7     Particles_universe[] univ;
8     public Universe(int particles)
9     {
10         number_universes++;
11         this.particles = particles;
12         univ = new Particles_universe[this.particles];
13     }
14     private void centerUniverse()
15     {
16         double sup_x=0, sup_y=0, sup_z=0;
17         for(int i = 0; i < number_particles_this_universe; i++)
18         {
19             sup_x += univ[i].getX();
20             sup_y += univ[i].getY();
21             sup_z += univ[i].getZ();
22         }
23         this.x = sup_x/number_particles_this_universe;
24         this.y = sup_y/number_particles_this_universe;
25         this.z = sup_z/number_particles_this_universe;
26     }
```

Листинг 4 — Нахождение расстояния между двумя вселенными (Класс Universe)(продолжение)

```
1      public void newParticlesUniverse(double x, double y, double z)
2      {
3          univ[number_particles_this_universe] = new Particles_universe(x,
4          y, z);
5          number_particles_this_universe++;
6          centerUniverse();
7      }
8
9      public double distanceBetweenUniverses(Universe universal)
10     {
11         return sqrt(pow(universal.getX()-x,2)+pow(universal.getY()-y,2)+
12         pow(universal.getZ()-z,2));
13     }
14
15     public double getX()
16     {
17         return this.x;
18     }
19
20     public double getY()
21     {
22         return this.y;
23     }
24
25     public double getZ()
26     {
27         return this.z;
28     }
29 }
```

Листинг 5 — Нахождение расстояния между двумя вселенными (Класс Particles)(продолжение)

```
1 public class Particles_universe {
2     private static long number_particles_universe;
3     private double x, y, z;
4     public Particles_universe(double x, double y, double z)
5     {
6         number_particles_universe++;
7         this.x = x;
8         this.y = y;
9         this.z = z;
10    }
11
12    public double getX()
13    {
14        return this.x;
15    }
16    public double getY()
17    {
18        return this.y;
19    }
20    public double getZ()
21    {
22        return this.z;
23    }
24    public long getN()
25    {
26        return number_particles_universe;
27    }
28 }
```

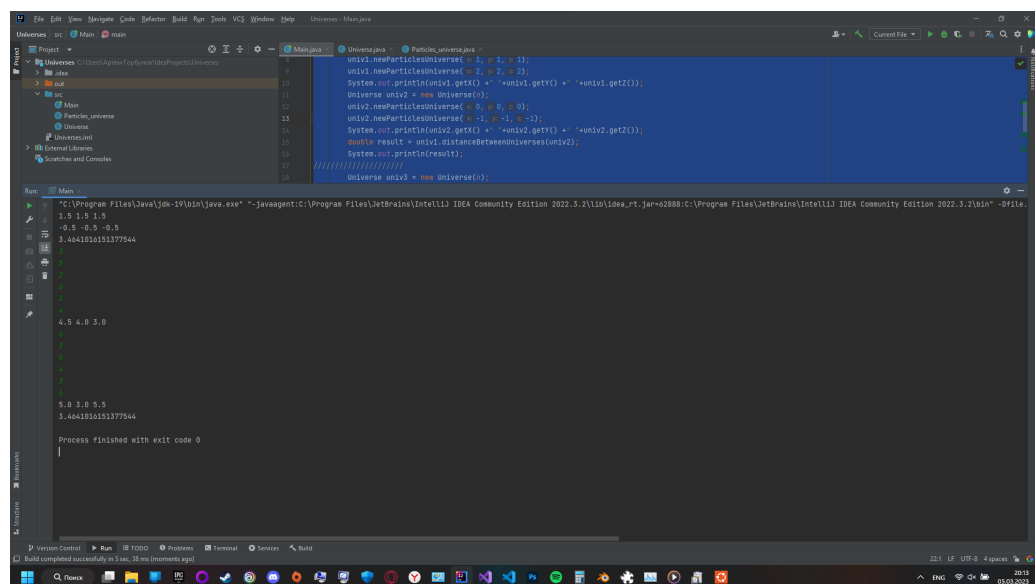


Рис. 1 — Вывод программы

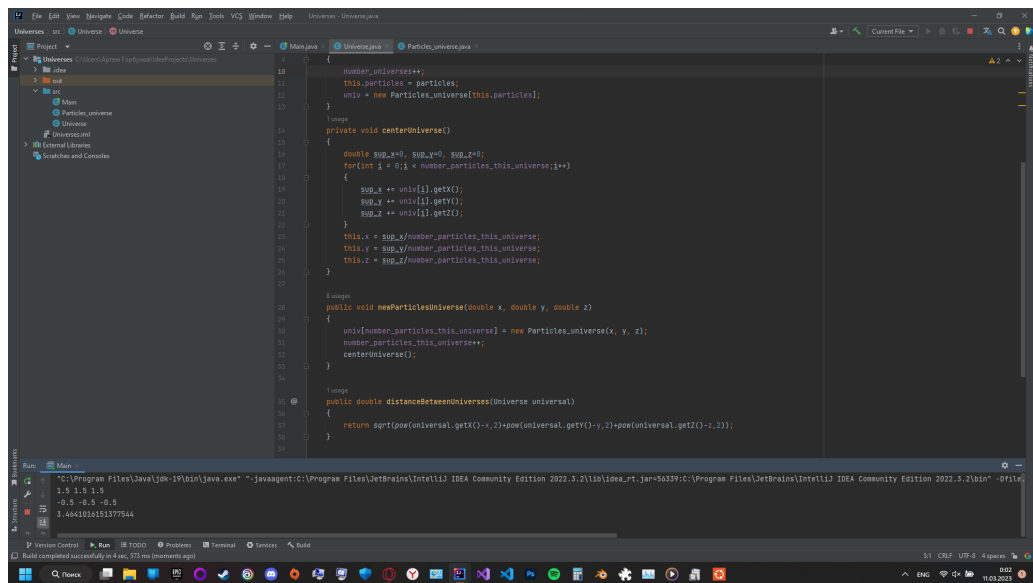


Рис. 2 — Работа программы(1)

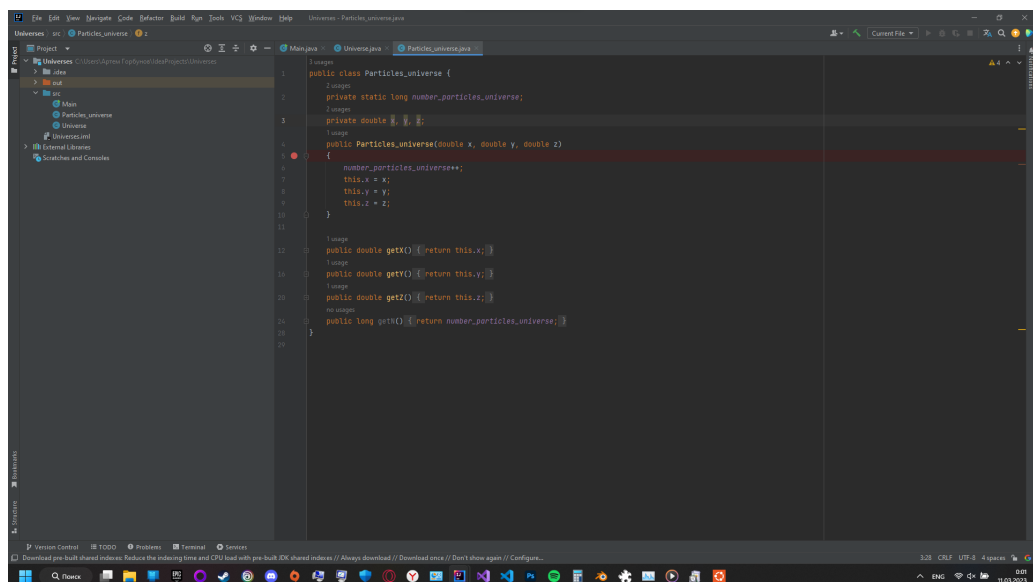


Рис. 3 — Работа программы(2)