



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. Вариант 35»

Студент группы ИУ9-22Б Жук 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.

2 Результаты

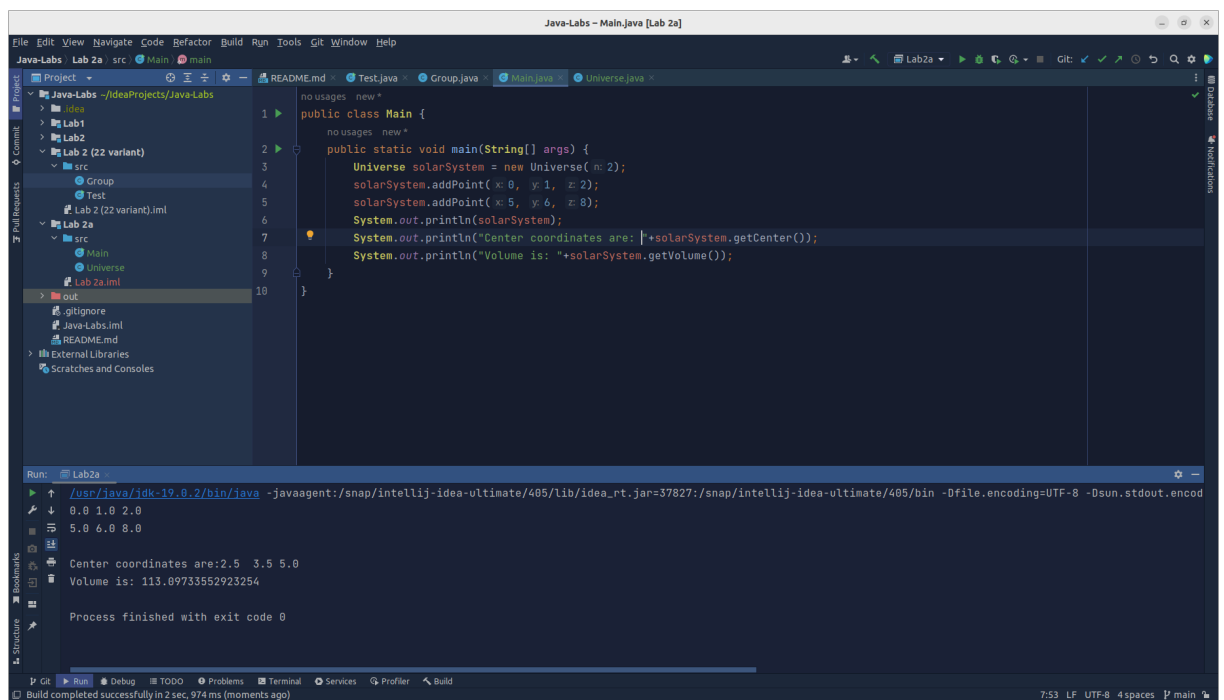


Рис. 1 — Скриншот работы программы

Для проверки правильности расчёта координат центра добавим ещё одну точку с координатами 1 5 25.

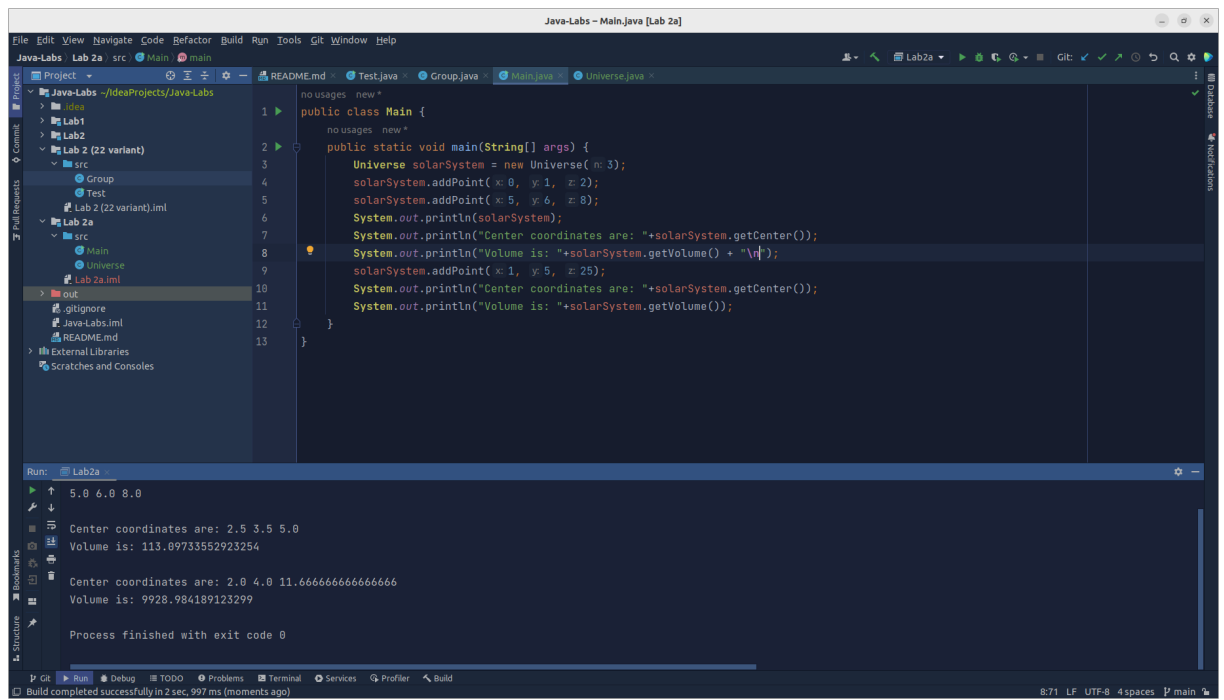


Рис. 2 — Скриншот перерасчёта объёма

Для проверки правильности работы с отрицательными числами добавим точку с координатами 37 -150 0.

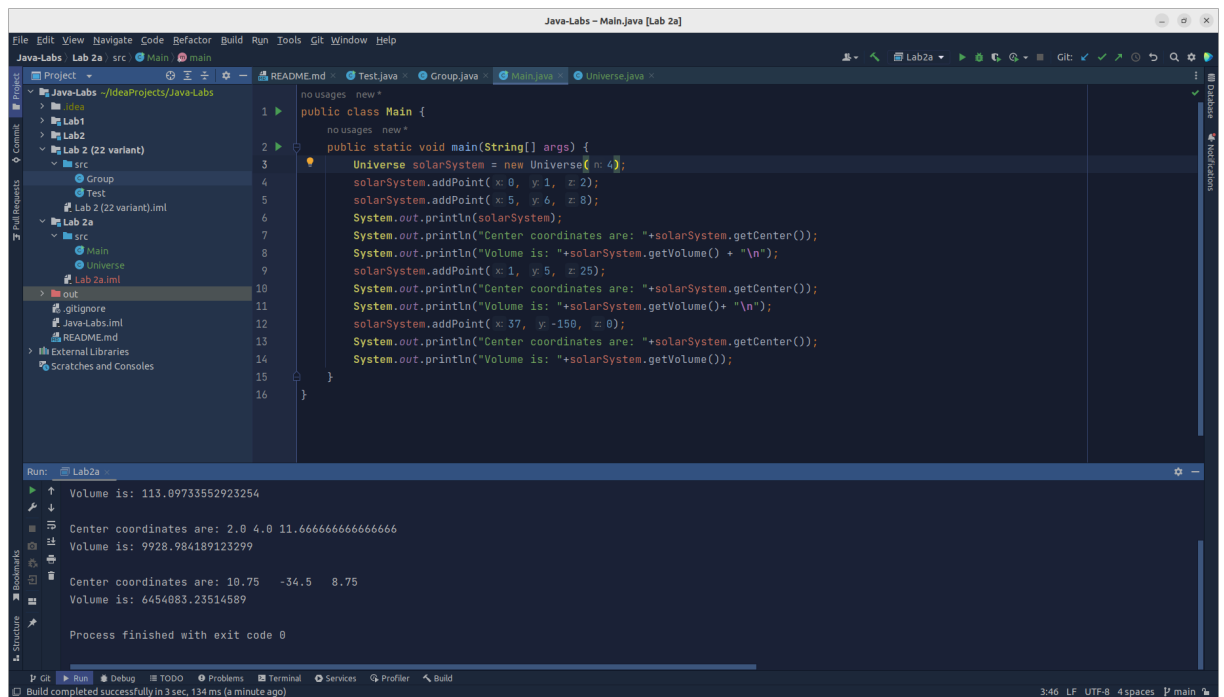


Рис. 3 — Скриншот перерасчёта

ЛИСТИНГ 1: Код класса Universe

```

1 public class Universe {
2     static int count = 0;
3     private Point[] points;
4     private Point center;
5     private Boolean centerRecalc;
6     Universe(int n){
7         this.points = new Point[n];
8         this.center = null;
9         this.centerRecalc = true;
10    }
11    private class Point{
12        double x, y, z;
13
14        Point(double x, double y, double z){
15            this.x = x;
16            this.y = y;
17            this.z = z;
18        }
19        public String toString(){
20            StringBuilder result = new StringBuilder();
21            result.append(this.x + "\t" + this.y + "\t" + this.z);
22            return result.toString();
23        }
24    }
25    public int getCount() {

```

```

26         return count;
27     }
28     public void addPoint(double x, double y, double z){
29         this.points[this.getCount()] = new Point(x, y, z);
30         count++;
31         this.centerRecalc = true;
32     }
33     private void calcCenter(){
34         double sumX = 0;
35         double sumY = 0;
36         double sumZ = 0;
37         for (int i = 0; i < count; i++){
38             sumX += this.points[i].x;
39         }
40         for (int i = 0; i < count; i++){
41             sumY += this.points[i].y;
42         }
43         for (int i = 0; i < count; i++){
44             sumZ += this.points[i].z;
45         }
46         this.center = new Point(sumX / this.getCount(), sumY / this.
getCount(), sumZ / this.getCount());
47         this.centerRecalc = false;
48     }
49
50     public Point getCenter(){
51         if (this.centerRecalc){
52             this.calcCenter();
53         }
54         Point res = new Point(this.center.x, this.center.y, this.center.
z);
55         return res;
56     }
57
58     public double getVolume(){
59         if (this.centerRecalc){
60             this.calcCenter();
61         }
62         double maxX = this.points[0].x;
63         double maxY = this.points[0].y;
64         double maxZ = this.points[0].z;
65         for (int i = 0; i < this.getCount(); i++){
66             if(Math.abs(this.points[i].x) > Math.abs(maxX) ){
67                 maxX = this.points[i].x;
68             }
69             if(Math.abs(this.points[i].y) > Math.abs(maxY)){

```

```

70         maxY = this.points[i].y;
71     }
72     if (Math.abs(this.points[i].z) > Math.abs(maxZ)) {
73         maxZ = this.points[i].z;
74     }
75 }
76 double r = Double.max( Math.abs(maxX- this.center.x), Math.abs(
maxY - this.center.y));
77 r = Double.max(r, Math.abs(maxZ - this.center.z));
78 double volume = ((double) 4 / (double) 3)* Math.PI * Math.pow(r,
3);
79 return volume;
80 }
81
82 public String toString(){
83     StringBuilder result = new StringBuilder(new StringBuilder());
84     for (int i = 0; i < count; i++){
85         result.append(this.points[i].x).append("\t").append(this.
points[i].y).append("\t").append(this.points[i].z).append("\n");
86     }
87     return result.toString();
88 }
89 }

```

ЛИСТИНГ 2 — Код класса Main

```

1 public class Main {
2     public static void main(String[] args) {
3         Universe solarSystem = new Universe(4);
4         solarSystem.addPoint(0, 1, 2);
5         solarSystem.addPoint(5, 6, 8);
6         System.out.println(solarSystem);
7         System.out.println("Center coordinates are: "+solarSystem.
getCenter());
8         System.out.println("Volume is: "+solarSystem.getVolume() + "\n")
;
9         solarSystem.addPoint(1, 5, 25);
10        System.out.println("Center coordinates are: "+solarSystem.
getCenter());
11        System.out.println("Volume is: "+solarSystem.getVolume()+ "\n");
12        solarSystem.addPoint(37, -150, 0);
13        System.out.println("Center coordinates are: "+solarSystem.
getCenter());
14        System.out.println("Volume is: "+solarSystem.getVolume());
15    }
16 }

```