

Lab 2: Class invariants

Problem B: Clash of the Planets

Submit your source code to Mooshak at:

<http://deei-mooshak.ualg.pt/~hdaniel>

up to Mar 15, 2020

Credits

(Figs. From:

<http://www.sci-news.com/astromony/water-world-exoplanets-life-06365.html>

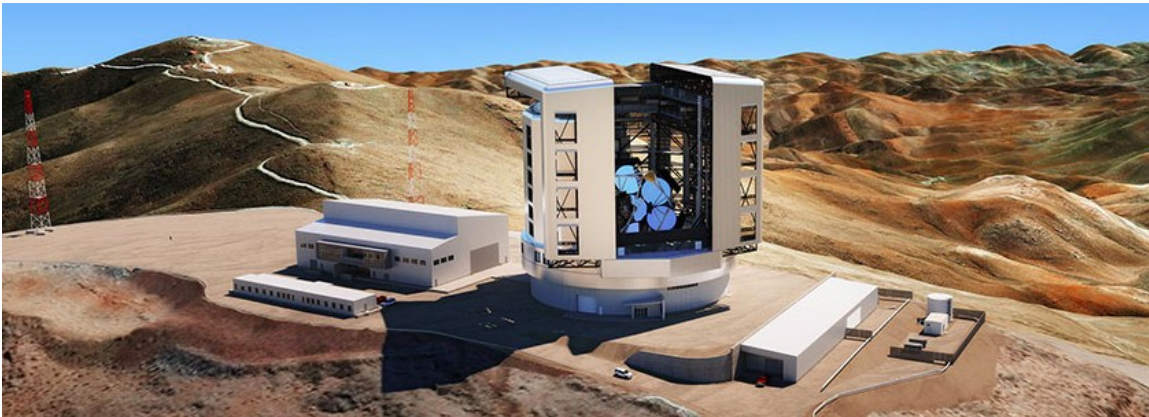
<https://www.gmto.org/2020/12/december-2020/>

<https://besthqwallpapers.com/space/clash-of-planets-explosion-apocalypse-galaxy-universe-55105>

)

Inspired by Time in T2 problem (OOP 2019/20)

The Giant Magellan Telescope (GMT) is being built in Andes. The GMT will have a resolving power 10 times greater than the Hubble Space Telescope. It will be able to really take another look on exoplanets in nearby star system.



When a new celestial object is found, its size, mass and rotation period are estimated.

Planets can be assumed roughly spheric, so its volume can be estimated knowing its radius only by:

$$V = \frac{4}{3} \pi r^3$$

Task

Develop a program that given a sphere with centre in a \mathbb{R}^3 point $C(x,y,z)$ and a real radius $r \geq 0$, computes its volume.

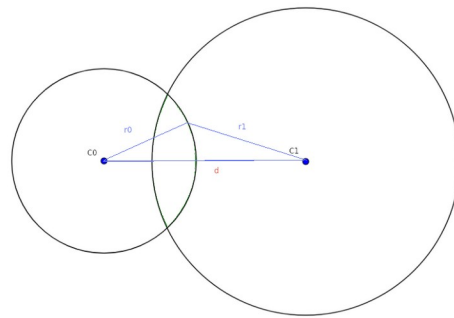
The program must also be able to detect if two spheres with centres A and B and radius r_A and r_B are colliding (intersecting).

The distance between 2 points in \mathbb{R}^3 is given by:

$$d = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2}$$

If the distance between the centre of the 2 spheres is less than the sum of both their radius, they are colliding:

$$\text{distance}(A, B) < r_A + r_B$$



Constraints

The program must be structured in three classes: **Point3D**, **Sphere** and **PlanetClash**, each in a different file.

Point3D class models points in \mathbb{R}^3 and implements a function to compute the distance between 2 of them.

Sphere class, models a sphere and the volume operation. For π use **Math.PI** constant. Class invariants must be enforced. Since it is a sphere in R^3 , the coordinates of the centre can be any real number, however the radius must be a real number equal or greater than zero. If radius is zero, the sphere is a point.

When constructing a sphere or when executing any operation that changes the value of the radius, the class invariant must be checked. If violated the program is aborted and printed an error message.

To abort a program can be used the `System.exit(1)` function (we will see other ways):

```
System.out.println("Class invariant violation: radius is < 0");  
System.exit(0); //must be zero or Mooshak will issue Runtime Error
```

Class **PlanetClash** reads input data from the console, uses **Sphere** class to compute volume and check if there is a collision, and writes output also to the console.

Input

The input has 2 lines. Each line have 4 real numbers to define a sphere, by this order: Center coordinates X, Y and Z and radius.

Or one line started with the string "col" and followed by 4 real numbers, for the centre coordinates X, Y and Z and radius of one sphere and more 4 real numbers, for the centre coordinates X, Y and Z and radius of one sphere and more 4

Output

The output have 3 lines. The first line presents the volume of the first sphere, with 2 decimal places of precision. The second line presents the volume of the second sphere and the third line presents the string "collision" or "no collision".

If any of the the spheres have a negative radius, thus violating class invariant, the program end immediately and outputs just one line with the string "iv: " followed by the definition of the Sphere that violated the class invariant, for instance:

```
iv: C(1.0,2.1,-3.5) r=-1.43
```

Input sample 0

```
0.0 0.0 0.0 3.0  
10.0 10.0 10.0 2.0
```

Output sample 0

```
113.10  
33.51  
no collision
```

Input sample 1

```
2.0 2.0 2.0 5.0  
4.0 4.0 4.0 2.0
```

Output sample 1

523.60
33.51
collision

Input sample 2

3.0 3.0 3.0 1.0
2.0 2.0 2.0 -1.0

Output sample 2

iv: C(2.0,2.0,2.0) r=-1.0

Input sample 3

3.0 3.0 3.0 -4.3
2.0 2.0 2.0 -1.0

Output sample 3

iv: C(3.0,3.0,3.0) r=-4.3