

# 几何体 系列

## 前情说明

本单元作业以 Task 的演进方式进行迭代，虽然我们直接给了同学们从 Task0 ~ Task6 的所有题面，以及仓库，但还是希望同学们能够从 Task0 开始顺序做题，并在做题的同时思考如何给后续的迭代留下修改的空间，而不至于同一个单元作业中每一个新的 Task 都要重新写一份代码。

本单元git数目较多，请注意这个问题并将代码push到正确的远端仓库中

## 总体目标

前面4个task (0-3) 有比较多的代码结构上的引导（题目要求具体到了实现某些方法），主要为了训练如何实现一系列基础的类。

Task4-5，加入了更多的类，主要面向类的结构、设计模式等。

Task6 加入了更难一些的操作，更多面向与各种代码技巧、Java自带容器的理解和使用等。

**希望同学们在自学java的同时，可以完成这些Task进行训练、巩固。在开始每个Task时，可以先阅读该Task内容，再根据自己对编程语言的了解，去搜索并深入学习该Task所需要的java语法、容器、特性。**

在开始前，你需要先学会java的基础语法，例如：表达式、for循环、if语句、输入输出、类，并学会编译运行java程序。

## Task 0

### 0.1

编写一个长方体盒子（名为 CuboidBox）的类，这个类需要有长(length)、宽(width)、高(height)三个属性。（在这道题中，这些属性可以被外部访问）。

示例代码：

```
1 public class CuboidBox {
2     // to be completed
3 }
4
5 import java.io.*;
6 public class MainClass {
7     public static void main(String[] args) {
8         Box mybox = new CuboidBox();
9         mybox.length = 10;
10        mybox.width = 10;
11        mybox.height = 20;
12        double vol;
13        vol = mybox.length * mybox.width * mybox.height;
14        System.out.println("The volume of the box is " + vol + ".");
15    }
16 }
```

期望输出: The volume of the box is 2000.0.

## 0.2

在主类中，像C语言的函数那样，编写一个“函数”，传入一个 `CuboidBox` 的对象，要求返回这个长方体盒子的体积。

示例代码：

```
1 struct cuboid_box
2 {
3     double length, width, height;
4 };
5
6 double calc_vol(struct cuboid_box box)
7 {
8     /* to be completed. */
9 }
```

```
1 import java.io.*;
2 public class MainClass {
3
4     private static double calcVol(CuboidBox box) {
5         // to be completed
6     }
7
8     public static void main(String[] args) {
9         Box mybox = new CuboidBox();
10        mybox.length = 10;
11        mybox.depth = 10;
12        mybox.height = 20;
13        System.out.println("The volume of the box is " + calcVol(mybox) +
14        ".");
15    }
```

期望输出: The volume of the box is 2000.0.

## 0.3

为 `CuboidBox` 类编写一个方法 `printVolume`，计算并输出该盒子的体积。

示例代码：

```

1 public class CuboidBox {
2     // to be completed
3 }
4
5 public class MainClass {
6     public static void main(String[] args) {
7         Box mybox = new CuboidBox();
8         mybox.length = 10;
9         mybox.width = 10;
10        mybox.height = 20;
11        mybox.printVolume();
12    }
13 }

```

期望输出: The volume of the box is 2000.0.

## 任务目标

**构建一个类，并尝试编写简单的函数。**

面向对象程序设计就是使用对象进行程序设计。对象代表现实世界中可以明确标识的一个实体。例如：一个学生、一个几何体、一个多项式都可以看做一个对象。每个对象都有自己独特的标识、属性和行为。

- 属性是由数据域及其当前值来表示的。例如，学生对象有一个数据域age，它是描述学生年龄的属性。
- 行为是由方法定义的，调用对象的一个方法就是要求对象完成一个动作。例如，可以为学生类定义getAge()方法，学生对象调用getAge()返回其年龄。

类是一个模板、蓝本或合约，用来定义对象的数据以及方法。对象是类的实例。创建实例的过程称为实例化。类和对象之间的关系类似于苹果派配方和苹果派的关系，可以用一种配方做出任意多的苹果派出来。

## Task 1

### 1.1

为 CuboidBox 类编写一个方法 getVolume，计算并返回该盒子的体积。

示例代码：

```

1 public class CuboidBox {
2     // to be completed
3 }
4
5 public class MainClass {
6     public static void main(String[] args) {
7         Box mybox = new CuboidBox();
8         mybox.length = 10;
9         mybox.width = 10;
10        mybox.height = 20;
11        System.out.println("The volume of the box is " + mybox.getVolume() +
12        ".");
13    }
14 }

```

期望输出: The volume of the box is 2000.0.

请考虑这种方法和传入写一个“函数”并传入对象的区别。

## 1.2

为 `CuboidBox` 类编写一个构造方法，构造方法中传入长宽高的参数，通过参数来初始化这个对象的数据。

示例代码：

```
1 public class CuboidBox {
2     // to be completed
3 }
4
5 import java.io.*;
6 public class MainClass {
7     public static void main(String[] args) {
8         Box mybox = new CuboidBox(10, 20, 10);
9         System.out.println("The volume of the box is " + mybox.getVolume() +
10         ".");
11     }
12 }
```

## 1.3

把 `CuboidBox` 封装起来！

要求：在外部，不能直接访问这个对象的各个属性，即：例如 `mybox.width = 12`；这样的赋值的访问是禁止的。同时，需要加入对这些属性的值的获取方法，例如，通过 `mybox.getWidth()` 来获取 `mybox` 的宽度。

示例代码：

```
1 public class CuboidBox {
2     // to be completed
3 }
4
5 import java.io.*;
6 public class MainClass {
7     public static void main(String[] args) {
8         Box mybox = new CuboidBox(10, 20, 10);
9         System.out.println("The length of the box is " + mybox.getLength() +
10         ".");
11         System.out.println("The volume of the box is " + mybox.getVolume() +
12         ".");
13     }
14 }
```

## 1.4

我们注意到，现在我们的长宽高这些属性，一旦确定就不可改变了（因为使用了 `private`）。因此，我们可以加入新的方法，使得这些属性可以改变。

构造一个 `ScaleCuboidBox`，拥有 `CuboidBox` 的全部数据和方法，并且能够通过 `set()` 方法来改变/设置长宽高。

## 最终效果

- 实现一个 `CuboidBox` 类
  - 有一个 `CuboidBox(double length, double width, double height)` 的构造函数。
  - 有 `double getLenth()` `double getWidth()` `double getHeight()` 三个方法，分别返回长、宽、高。
  - 有 `double getVolume()` 方法，返回该长方体的体积。
- 实现一个 `ScaleCuboidBox` 类
  - 包含 `CuboidBox` 的所有要求。
  - 有 `setLenth(double length)` `setWidth(double width)` `setHeight(double height)` 三个方法，分别设定或者改变长、宽、高。

## 评测方式

### 题目描述

一开始给出一个长方体的长宽高。

之后，有7种操作：

1. 查询长
2. 查询宽
3. 查询高
4. 更改长
5. 更改宽
6. 更改高
7. 询问体积

### 输入输出格式

第一行给出 `length width height` 三个浮点数，分别表示长宽高。

第二行一个数 `n`，表示操作数

接下来 `n` 行，每行有一个操作类型数，分别对应上方的七种操作。

如果为4、5、6，则在操作类型数后会有一个数，表示需要更改成的值。

对于查询操作，每个查询操作输出一行，输出对应的值。

### 样例输入

```
1 1 2 3
2 12
3 1
4 4 4
5 1
6 7
7 2
8 4 5
9 2
10 7
11 3
12 5 6
13 3
14 7
```

## 样例输出

```
1 1.0
2 4.0
3 24.0
4 2.0
5 5.0
6 60.0
7 3.0
8 6.0
9 120.0
```

## 评测对比方法

你的输出数值需要和正确数值相等。

如果你的答案和正确数值之间的差的绝对值小于等于 $10^{-5}$ 则认为是相等的。

## 任务目标

### 学会构建构造方法和对类进行封装，理解封装的作用

java类使用变量定义数据域，使用方法定义行为。除此之外，类还提供了一种称为构造方法的特殊类型的方法，调用它可以创建一个内新对象。构造方法本身是可以完成任何动作的，但是设计构造方法是为了完成初始化动作，例如初始化对象的数据域。

封装是指隐藏对象的属性和实现细节，仅对外提供公共访问方式。这样做的优点是提高复用性、提高安全性和隐藏了实现细节。

关键字private，被private修饰的成员只能在本类中使用。

对属性进行安全性的封装的一般实现步骤

1. 对属性进行私有化。
2. 对外提供公开的set/get方法。

## Task 2

### 基本要求

利用长方体类进行计算

完成长方体 Cubiod 类，至少包含以下方法：

- 有一个 `CuboidBox(double length, double width, double height)` 的构造函数
- 有 `double getLenth()` `double getWidth()` `double getHeight()` 三个方法，分别返回长、宽、高。
- 有 `setLenth(double length)` `setWidth(double width)` `setHeight(double height)` 三个方法，分别设定或者改变长、宽、高。
- 有 `double getVolume()` 方法，返回该长方体的体积。

### 题目描述

有n个长方体的**可重集合**，初始均为空，编号为1-n。

有以下几种操作：

1. 询问某个集合中体积最大的长方体的体积

- 2. 询问某个集合中的长方体的体积和
- 3. 新增一个长方体至某个集合。

保证询问的集合均不为空

## 输入格式

### 描述方法

长方体以以下形式描述：

a b c 三个数字，表示长 宽 高。

操作以以下形式描述：

type attribute

其中，type对应上文中的操作，不同操作的具体描述如下：（attribute均无换行）

type	attribute	意义	输出
1	i	询问第i个集合中体积最大的长方体的体积	体积，一个数
2	i	询问第i个集合中的长方体的体积和	体积和，一个数
3	i description	在第i个集合中添加描述为description的长方体，其中，长方的描述参见上方。	无输出

### 格式

输入中，第一行两个数n, m，表示集合的个数和操作的个数。

接下来m行，每行一个操作，如上文描述。

数值均可能包含小数。

## 输出格式

对于每个需要输出的操作，输出一行。

## 数据范围

- $1 \leq n, m \leq 20000$
- 对几何体的描述均有是意义的

## 评测方法

输出数值时，你的输出数值需要和正确数值相等。

如果你的输出值和正确数值之间的差的绝对值小于等于 $10^{-5}$ 则认为是相等的。

## 样例输入

```
1 | 3 4
2 | 3 1 1 2 3
3 | 3 1 2 3 4
4 | 2 1
5 | 1 1
```

## 样例输出

```
1 | 30.0
2 | 24.0
```

## 任务目标

**建立一个对象的集合，实现向集合中增加对象和访问集合中对象的操作**

熟悉对容器的操作，题目中要求使用的时可重复集合，可以考虑使用静态数组或者ArrayList（推荐使用ArrayList），并思考如果是不可重复集合怎么做。（HashMap、HashSet一类的数据结构，可以快速管理和调配手中无序的数据。）

## 体会与感想

在task2中我们实现了通过容器来对对象进行管理，思考一下如果直接让你从task2开始做和从task0开始做会有什么区别。

## Task 3

### 基本要求

实现一个3维向量类。

- 有便于使用的构造函数
- 有模长运算
- 有一系列对对象属性的查询函数
- 可以实现向量的加、减、点乘、叉乘。

可以自定义实现方法。

### 题目描述

有一个包含n个三维向量的几何，分别为向量 $v_1, v_2, \dots, v_n$ 。

有以下6种操作：

1. 给出 $i, j$ ，求 $v_i + v_j$ ，输出结果。假设当前集合中有 $cnt$ 个向量，则将该向量设为 $v_{cnt+1}$ ，并加入集合。
2. 给出 $i, j$ ，求 $v_i - v_j$ ，输出结果。假设当前集合中有 $cnt$ 个向量，则将该向量设为 $v_{cnt+1}$ ，并加入集合。
3. 给出 $i, j$ ，求 $v_i \cdot v_j$ ，输出结果。
4. 给出 $i, j$ ，求 $v_i \times v_j$ ，输出结果。假设当前集合中有 $cnt$ 个向量，则将该向量设为 $v_{cnt+1}$ ，并加入集合。
5. 输出当前集合中，模长最大的向量的模长
6. 输出当前集合中，模长最小的向量的模长

### 输入描述



第一行包含一个数  $n$ ，表示初始向量个数。

接下来  $n$  行，每行三个数，表示一个向量

接下来一行一个数  $q$ ，表示操作个数。

接下来  $q$  行，每行一个第一个数为操作编号，对应上方六种操作。如果为 1、2、3、4 号操作，则还有两个数  $i, j$ 。

**数值均可能包含小数。**

## 输出描述

对于每个操作，输出一个结果。

向量输出格式为

```
x y z
```

## 数据范围

$1 \leq n, q, \leq 100000$

## 评测方法

你的输出数值需要和正确数值相等。

如果你的答案和正确数值之间的差的绝对值小于等于  $10^{-5}$  则认为是相等的。

## 样例输入

```
1 2
2 1.0 2.0 3.0
3 1.0 1.0 1.0
4 6
5 1 1 2
6 2 1 2
7 3 1 2
8 4 1 2
9 5
10 6
```

## 样例输出

```
1 2.0 3.0 4.0
2 0.0 1.0 2.0
3 6.0
4 -1.0 2.0 -1.0
5 5.385164807134504
6 1.7320508075688772
```

## 任务目标

**进一步熟悉类的构建和基本容器的使用**

题目中要求了求集合内最大和最小，请搜索并思考使用何种容器实现较好。

## 体会与感想

在task3中希望你自己从头开始建立一个类，通过前面几个task的学习，相信聪明的你应该很容易就能完成。那么对比三维向量类和几何体类，请你分析它们有什么共同之处，又有什么区别，并总结一下构建类的一般方法。

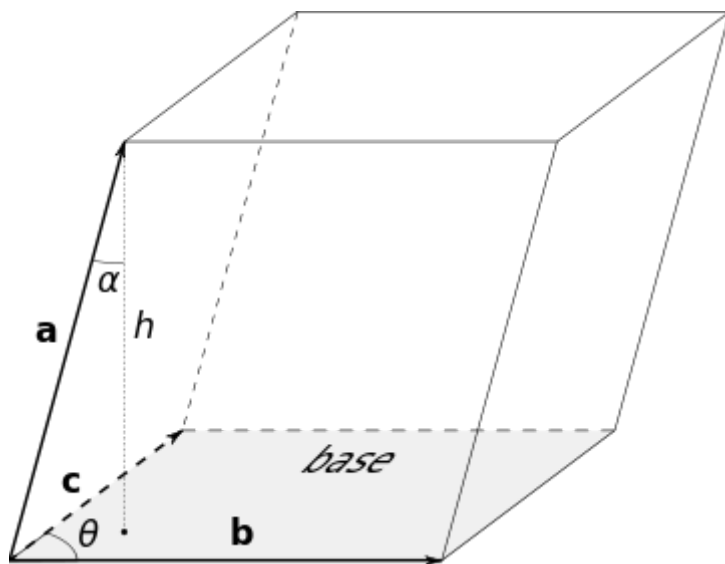
## Task 4

### 基本要求

在几何学中，平行六面体是由六个平行四边形所组成的三维立体，是一种平行多面体。平行六面体的三个等价的定义为：

六个面都是平行四边形的多面体；有三对对面平行的六面体；底面为平行四边形的棱柱。长方体（六个面都是长方形）、正方体（六个面都是正方形），以及菱面体（六个面都是菱形）都是平行六面体的特殊情况。

——[维基百科.平行六面体](#)



完成一个立方体 Cube 类，至少包含以下方法：

- 有一个 `CuboidBox(double length)` 的构造函数，传入立方体的棱长
- 有 `double getLenth()` 三个方法，返回棱长。
- 有 `setLenth(double length)` 方法，设定或者改变立方体的棱长。
- 有 `double getVolume()` 方法，返回该立方体方体的体积。

建议通过继承的方式，使得 Cube 继承自 Cubiod。

自行设计平行六面体类，至少满足以下要求。

- 用于描述平行六面体的属性可以自定义。
- 有构造函数
- 有修改方法、查询方法。
- 可以求体积。
- 改造之前的长方体类，使其继承自该类。

提示：可以使用向量描述平行六面体

### 题目描述

有n个几何体的**可重集合**，初始为空，编号为1-n，几何体包含立方体，长方体和平行六面体。

有以下几种操作：

- 1. 询问某个集合中体积最大的几何体的体积
- 2. 询问某个集合中体积最大的几何体的类型
- 3. 询问某个集合中的几何体的体积和
- 4. 新增一个几何体至某个集合。

保证询问的集合均不为空

输入格式

描述方法

几何体以以下形式描述：

type attribute

其中，不同几何体格式如下：（attribute中的字母代表数）

名字	type	attribute	意义
平行六面体	1	ax bx by cx cy cz	$(ax, 0, 0)(bx, by, 0)(cx, cy, cz)$ 三个向量表示的平行六面体
长方体	1.1	a b c	长 宽 高
正方体	1.1.1	a	棱长

操作以以下形式描述：

type attribute

其中，type对应上文中的操作，不同操作的具体描述如下：（attribute均无换行）

type	attribute	意义	输出
1	i	询问第i个集合中体积最大的几何体的体积	体积，一个数
2	i	询问第i个集合中体积最大的几何体的类型	类型，输出几何体的type，如上表描述
3	i	询问第i个集合中的几何体的体积和	体积和，一个数
4	i description	在第i个集合中添加描述为description的几何体，其中，几何体的描述参见上方。	无输出

格式

输入中，第一行两个数n, m，表示集合的个数和操作的个数。

接下来m行，每行一个操作，如上文描述。

数值均可能包含小数。

输出格式

对于每个需要输出的操作，输出一行。

## 数据范围

- $1 \leq n, m \leq 20000$
- 对几何体的描述均有是意义的

## 样例输入

```
1 2 5
2 4 1 1 1 2 3 4 5 6
3 4 1 1.1.1 6
4 1 1
5 2 1
6 3 1
```

## 样例输出

```
1 216.0
2 1.1.1
3 234.0
```

## 评测方法

输出数值时，你的输出数值需要和正确数值相等。

如果你的输出值和正确数值之间的差的绝对值小于等于 $10^{-5}$ 则认为是相等的。

## 任务目标

### 学习类的继承关系

继承使得你可以定义一个通用的类（即父类），之后继承该类为一个特定的类（即子类）。

使用类来对同一对象建模时，不同的类可能会有一些共同的属性和行为，可以在一个通用类中表达这些共同之处，并被其他类所共享。

例如，考虑老师和学生这两个对象，他们有许多共同的属性和行为。他们都有性别，年龄等特征。可以建立一个People类来建模所有的人，这个类包括属性sex和age，以及使用于这些属性的get/set方法。Student类和Teacher类可以继承自People类，并加上自己独有的方法。

在Java中，如果类C1继承自另一个类C2，那么就将C1称为子类，将C2称为父类。子类从它的父类中继承可访问的数据域和方法。

## 体会与感想

task4中学习了面向对象中很重要的一部分——继承。通过完成这一部分的代码，请你分析一下，使用继承会为你代码的编写带来怎样的方便之处（可以从写代码的过程，代码的层次等方面进行分析）。

## Task 5

### 基本要求

实现平行六面体、长方体、立方体、圆台体、圆柱体、圆锥体、球体类

- 这些类应该符合某种继承关系
- 应该均可求体积

提示：

- 使用多态，建立一个共同的父类/接口
- 每个类均有求体积的方法，生成描述信息的方法...
- 建造一个类，这个类可以根据输入自动生成对应的几何体对象（工厂模式）

## 题目描述

有n个几何体的**可重集合**，初始为空，编号为1-n，几何体包含行六面体、长方体、立方体、圆台体、圆柱体、圆锥体、球体。

有以下几种操作：

1. 询问某个集合中体积最大的几何体的体积
2. 询问某个集合中体积最大的几何体的信息
3. 询问某个集合中的几何体的体积和
4. 新增一个几何体至某个集合。

保证询问的集合均不为空

## 输入格式

### 几何体描述方法

几何体以以下形式描述：

`type attribute`

其中，不同几何体的格式如下：（attribute中的字母代表对应的数）

名字	type	attribute	意义
平行六面体	1	<code>ax bx by cx cy cz</code>	$(ax, 0, 0)(bx, by, 0)(cx, cy, cz)$ 三个向量表示的平行六面体
长方体	1.1	<code>a b c</code>	长 宽 高
正方体	1.1.1	<code>a</code>	棱长
圆台体	2	<code>r1 r2 h</code>	底面半径 顶面半径 高度
圆柱体	2.1	<code>r h</code>	半径 高
圆锥体	2.2	<code>r h</code>	半径 高
球体	3	<code>r</code>	半径

### 操作描述方法

操作以以下形式描述：

`type attribute`

其中，type对应上文中的操作，不同操作的具体描述如下：（attribute均无换行）

type	attribute	意义	输出
1	i	询问第i个集合中体积最大的几何体的体积	体积，一个数
2	i	询问第i个集合中体积最大的几何体的信息	输出信息参见几何体描述方法（即：按照描述输出）
3	i	询问第i个集合中的几何体的体积和	体积和，一个数
4	i description	在第i个集合中添加描述为description的几何体，其中，几何体的描述参见上方。	无输出

### 格式

输入中，第一行两个数n, m，表示集合的个数和操作的个数。

接下来m行，每行一个操作，如上文描述。

数值均可能包含小数。

### 输出格式

对于每个需要输出的操作，输出一行。

### 数据范围

- $1 \leq n, m \leq 20000$
- 对几何体的描述均有是意义的

### 评测方法

输出数值时，你的输出数值需要和正确数值相等。

如果你的输出值和正确数值之间的差的绝对值小于等于 $10^{-5}$ 则认为是相等的。

### 样例输入

```
1 2 6
2 4 1 1 1 2 3 4 5 6
3 4 1 2.1 1 3
4 4 1 3 5
5 1 1
6 2 1
7 3 1
```

### 样例输出

```
1 392.6990816987241
2 3 5.0
3 420.1238596594935
```

## 任务目标

### 学习使用多态，了解设计模式中的工厂模式

多态意味着父类型的变量可以引用子类型的变量。

面向对象程序设计的三大支柱是封装、继承和多态。这一部分将介绍多态。继承关系使一个子类能继承父类的特征，并且附加一些新的特征。子类是它父类的特殊化，每个子类的实例都是其父类的实例，但反过来不成立。使用父类对象的地方都可以使用子类对象替换。

工厂模式，最终解决的问题就是项目里面对象实例化的问题，要大量实例化某些类的对象，就要用到工厂模式。因为如果把这些实例化过程放在项目里面，它的变动、扩展、升级就会很麻烦，所以应将实例化放在某个区域一起批量管理。

## 体会与感想

task5中学习了面向对象中很重要的一部分——多态。通过完成这一部分的代码，请你分析一下，多态意义与优点。并思考一下如果没有多态会带来怎样的麻烦。

## Task 6

### 基本要求

实现平行六面体、长方体、立方体、圆台体、圆柱体、圆锥体、球体类。

- 这些类应该符合某种继承关系
- 应该均可求体积
- 可以使用比较函数比较（详细见下方对集合重复的定义）

提示：

- 使用多态，建立一个共同的父类/接口
- 建造一个类，这个类可以根据输入自动生成对应的几何体对象（工厂模式）

### 题目描述

有n个几何体的**不可重集合**，初始为空，编号为1-n，几何体包含行六面体、长方体、立方体、圆台体、圆柱体、圆锥体、球体。

有以下几种操作：

1. 询问某个集合中体积最大的几何体的体积
2. 询问某个集合中体积最大的几何体的信息
3. 询问某个集合中的几何体的体积和
4. 新增一个几何体至某个集合。
5. 对两个集合求并集，假设当前已经有cnt个集合，则将得到的集合作为编号为cnt+1的集合。

**不保证询问的集合均不为空**

**关于几何体相同的定义：**

- 两个集合在输入的时候，类型应该相同（例如：即使是长宽高都相同的长方体，也认为和立方体不同）

- 两个集合在输入的时候，各项各项参数大小应该对应相同（例如：长宽高为 1 2 3 的长方体和长宽高为 2 1 3 的长方体 我们认为是不一样的，但是棱长1和棱长1.0的正方体认为是相同的）

如果1、2操作询问的集合为空，则输出报错信息，信息详见输出格式

## 输入格式

### 几何体描述方法

几何体以以下形式描述：

`type attribute`

其中，不同几何体格式如下：（attribute中的字母代表对应的数）

名字	type	attribute	意义
平行六面体	1	<code>ax bx by cx cy cz</code>	$(ax, 0, 0)(bx, by, 0)(cx, cy, cz)$ 三个向量表示的平行六面体
长方体	1.1	<code>a b c</code>	长 宽 高
正方体	1.1.1	<code>a</code>	棱长
圆台体	2	<code>r1 r2 h</code>	底面半径 顶面半径 高度
圆柱体	2.1	<code>r h</code>	半径 高
圆锥体	2.2	<code>r h</code>	半径 高
球体	3	<code>r</code>	半径

操作以以下形式描述：

`type attribute`

其中，type对应上文中的操作，不同操作的具体描述如下：（attribute均无换行）

type	attribute	意义	输出
1	<code>i</code>	询问第i个集合中体积最大的几何体的体积	体积，一个数 或 报错
2	<code>i</code>	询问第i个集合中体积最大的几何体的信息	输出信息参见几何体描述方法 或者 报错
3	<code>i</code>	询问第i个集合中的几何体的体积和	体积和，一个数
4	<code>i description</code>	在第i个集合中添加描述为description的几何体，其中，几何体的描述参见上方。	无输出
5	<code>i j</code>	对i,j集合求并集，假设当前已经有cnt个集合，则将得到的集合作为编号为cnt+1的集合。	无输出

## 格式



输入中，第一行两个数 $n, m$ ，表示集合的个数和操作的个数。

接下来 $m$ 行，每行一个操作，如上文描述。

数值均可能包含小数。

## 输出格式

对于每个需要输出的操作，输出一行。

报错请输出

```
Sorry, the set is empty!
```

## 数据范围

- $1 \leq n \leq 5000$
- 对几何体的描述均有意义的
- 输入数据最多包含6位小数
- 保证最大集合的大小小于等于5000

## 评测方法

输出数值时，你的输出数值需要和正确数值相等。

在进行答案比较时，如果你的输出值和正确数值之间的差的绝对值小于等于 $10^{-5}$ 则认为是相等的。

## 样例输入

```
1 2 8
2 4 1 1.1 1 1
3 4 1 2.1 1 3
4 4 2 1.1 1 1
5 1 1
6 2 1
7 3 1
8 5 1 2
9 3 3
```

## 样例输出

```
1 9.42477796076938
2 2.1 1.0 3.0
3 10.42477796076938
4 10.42477796076938
```

## 提示

判断几何体相同时，需要参数对应完全“相等”，这里的“相等”和答案比较的“相等”不同，需要两个几何体，在输入时，对应参数在数学意义上的完全相等，例如 $2.499 \neq 2.5$ 。

## 任务目标

加入了判断对象是否相同的操作，可以通过建立比较函数`equals()`来实现

比较对象是否相同，只需比较这个对象中的所有属性是否相同。

可以考虑使用HashMap, HashSet等容器。

## 体会与感想

在task6中，我们需要比较两个对象是否相同，请你大致描述一下你的比较方法，并思考这种方法的优缺点，查找资料看看是否有更好的方法。

了解HashMap, HashSet等容器的内部实现并思考一下这些容器与ArrayList在性能上的差异。