Java Day 06

数组拷贝

已分配空间的两个数组间进行数据内容拷贝

System.arraycopy(…)

多维数组：数组的每个成员又是一个数组

公司年销售额求和-->SumMultiArray.java

某公司按照季度和月份统计的数据如下：单位(万元)

int[4][3] array

第一季度：22,66,44

第二季度：77,33,88

第三季度：25,45,65

第四季度：11,66,99

可变参数

底层用数组实现，一个参数列表中只能出现一个可变参数，并在所有参数的最后声明-->VarargsTest.java

```
sum(int ...a){
   …
}
sum(1,2,3);
```

面向对象编程思想

强调的是通过调用对象的行为来实现功能，而不是自己一步一步的去操作实现。

例如洗衣服：

面向过程：把衣服脱下来-->找一个盆-->放点洗衣粉-->加点水-->浸泡 10 分钟-->揉一揉-->清洗衣服-->拧干-->晾起来

面向对象：把衣服脱下来-->打开全自动洗衣机-->扔衣服-->按钮-->晾起来

两者区别：

面向过程：强调步骤。

面向对象：强调对象，这里的对象就是洗衣机。

面向对象思想是一种更符合我们思考习惯的思想，它可以将复杂的事情简单化，并将我们从执行者变成了指挥者。

面向对象特性：

封装(encapsulation)

继承(inheritance)

多态(polymorphism)

类和对象

类是现实世界中，具有相同属性和行为的事物的抽象，可以看成是一类事物的模板。

属性：就是该事物的状态信息。

行为：就是该事物能够做什么。

例如：人类、鸟类、鱼类

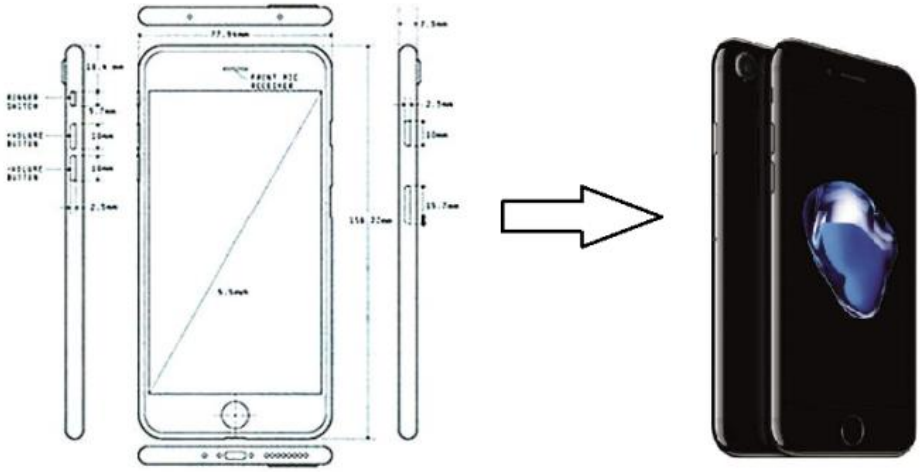
对象是类的具体化，是类的一个实例。

类与对象的关系：

类是对一类事物的描述，是抽象的。

对象是一类事物的实例，是具体的。

类是对象的模板，对象是类的实体。



手机的设计图（抽象的）　　真正的手机（具体的）

面向对象编程步骤

定义类

```
public class 类名 {
    //成员变量
    //成员方法
}
```

创建对象

```
类名 对象名 = new 类名();
```

访问成员

```
对象名.成员变量;
```

对象名.成员方法();

方法概述

什么是方法

完成特定功能的代码块

为什么要有方法

提高代码的复用性

方法的格式

修饰符 返回值类型 方法名(参数类型 参数名 1,参数类型 参数名 2...) {

```
    方法体语句;
    return 返回值;
}
```

方法的格式说明

修饰符：目前就用 public static，后面再详细讲解其他的修饰符

返回值类型：就是功能结果的数据类型

方法名：符合命名规则即可，方便我们的调用

参数：

实际参数：就是方法调用时的，实际参与运算的。

形式参数：就是方法定义上的，用于接收实际参数的。

参数类型：就是参数的数据类型

参数名：就是变量名

方法体语句：就是完成功能的代码

`return`：结束方法的，可以携带返回值也可以不携带

返回值：就是功能的结果，由 `return` 带给调用者。

方法的注意事项

方法不调用不执行

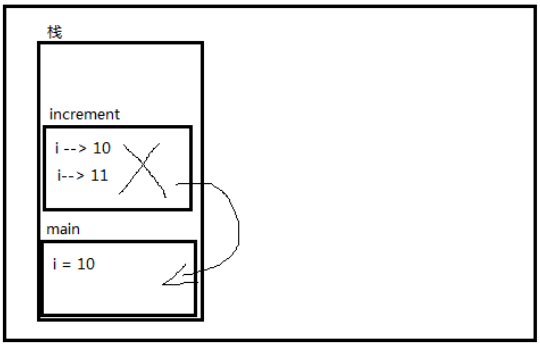
方法与方法是平级关系，不能嵌套定义

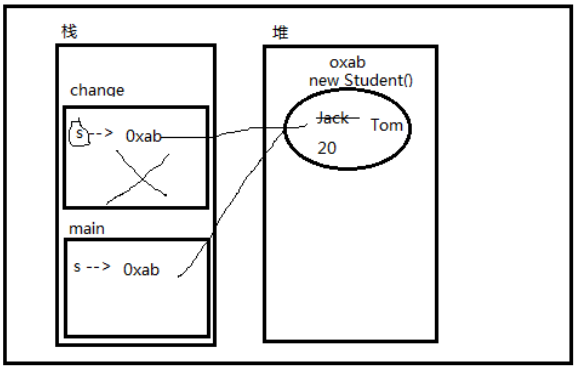
方法定义的时候参数之间用逗号隔开

方法调用的时候不用再传递数据类型

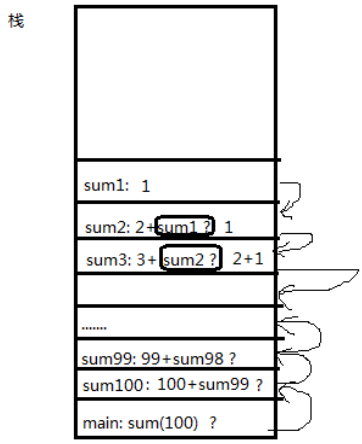
如果方法有明确的返回值，一定要有 `return` 带回一个值

方法调用时，基本数据类型传值，类类型传引用(地址)





递归：方法自己调用自己-->Recursion.java



好处：不知道循环次数

弊端：不能调用次数过多，可能导致栈溢出

例如：

1）斐波那契数列：1,1,2,3,5,8,13,21

2）古典问题：3 个月起每个月都生一对兔子，小兔子长到第三个月后每个月又生一对兔子，假如兔子都不死，问每个月的兔子总数为多少?

那么我们假定第一个月的兔子为小兔子，第二个月为中兔子，第三个月之后就为大兔子，那么第一个月分别有 1、0、0，第二个月分别为 0、1、0，第三个月分别为 1、0、1，第四个月分别为,1、1、1，第五个月分别为 2、1、2，第六个月分别为 3、2、3，第七个月分别为 5、3、5.....

兔子总数分别为：1、1、2、3、5、8、13.....

3）阶乘：n! --> 5!=5*4*3*2*1

封装

封装概念

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。

封装好处

隐藏实现细节，提供公共的访问方式

提高了代码的复用性

提高安全性

封装原则

将不需要对外提供的内容都隐藏起来

把属性隐藏，提供公共方法对其访问

封装的操作：private 关键字

private 关键字特点

是一个权限修饰符

可以修饰成员变量和成员方法

被其修饰的成员只能在本类中被访问

private 仅仅是封装的一种体现形式,不能说封装就是私有

人类年龄赋值的问题

pojo,java bean

---

this

代表所在类的当前对象的引用（地址值）

this.成员变量名;

this.成员方法名;

方法重载-->OverloadingTest.java

在同一类中

相同的方法名

参数列表有所不同（参数类型、参数数量、参数顺序）

不关心返回值类型