复习：

1. 数组的应用：

数组排序 --- 选择排序 ---选定一位，然后依次和其他位上的元素来进行比较 --- O(n2), o(1)

数组反转 --- 首尾交换

数组元素的查找 --- 数组无序---遍历，数组有序---二分查找 --- O(logn), o(1)

数组的复制 --- System.arraycopy Arrays.copyOf

2. 二维数组：

用于存储一维数组的容器

二维数组的内存存储

## 方法

为了使某一段代码重复使用，将这段代码提取到一个新的结构---方法/函数。

方法在定义的时候，需要明确返回值类型以及参数。

明确返回值类型---根据方法的执行结果的类型来确定 --- 如果方法执行之后没有结果，返回值类型就定义为void

明确参数类型---判断方法在执行的过程是否有未知量来参与运算

定义方法的时候写在()的参数称之为形参，调用方法的时候传入的数据称之为实参。

### 方法的重载

在同一类中定义多个方法名一致而参数列表不同的方法 --- 方法的重载

所谓参数列表不同实际上是看的每一位上的参数类型是否不同或者参数个数是否不同

Arrays.toString(int[] arr);

Arrays.toString(double[] arr);

Arrays.sort

### 方法的递归

在方法中调用自己本身

练习：求n! --- n的阶乘

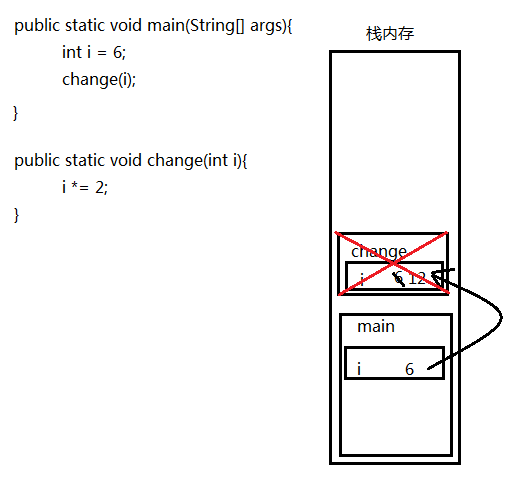
5! = 5 \* 4 \* 3 \* 2 \* 1 = 120

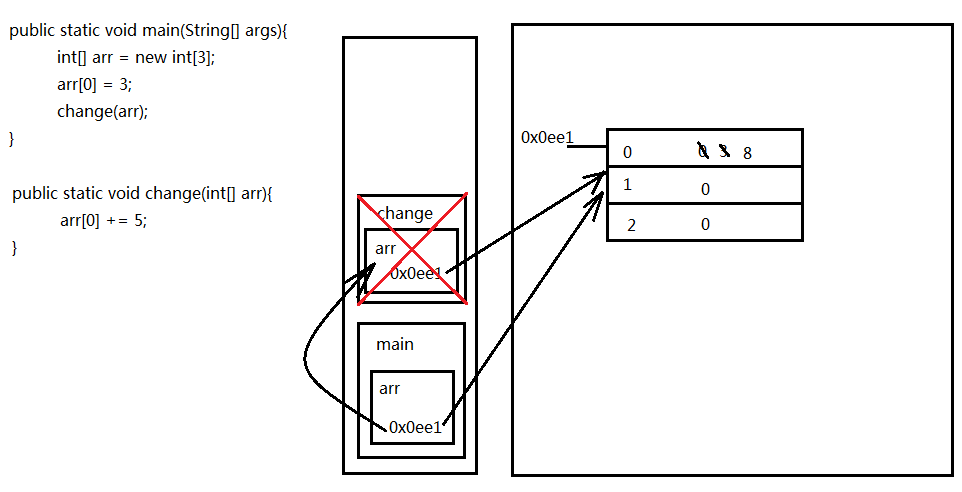
n! = n \* (n - 1)!

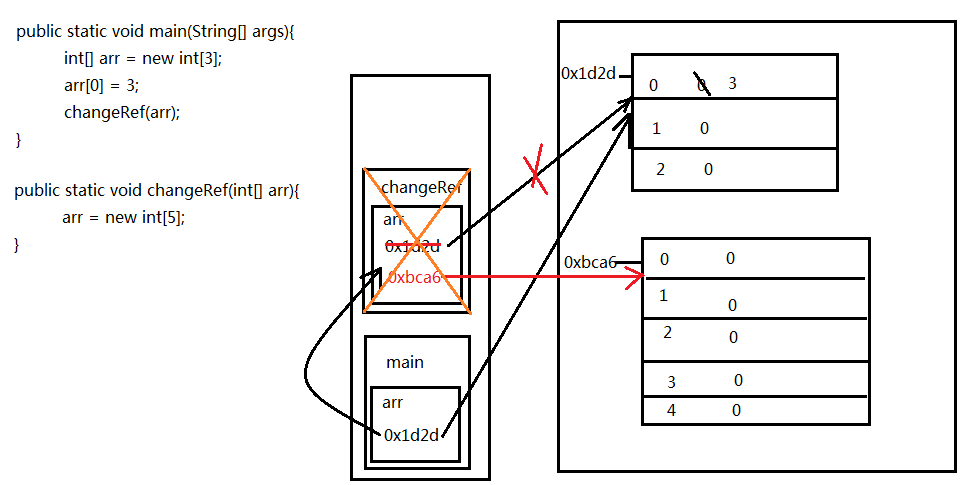
StackOverFlowError --- 栈溢出错误 --- 方法在栈内存中执行 --- 递归在使用的时候不能递归层数太深，递归过深导致栈溢出错误

从空间上考虑，递归比循环要更加耗费内存；从时间考虑，递归比循环要花费更多的时间。 --- 递归的效率要低于循环

### 方法的传值问题







IDE---智能开发工具

Eclipse MyEclipse Intelli J---IDEA ...

Eclipse --- 日食 --- 免费、绿色、基于插件的

Luna --- Mars --- Neon --- Oxygen

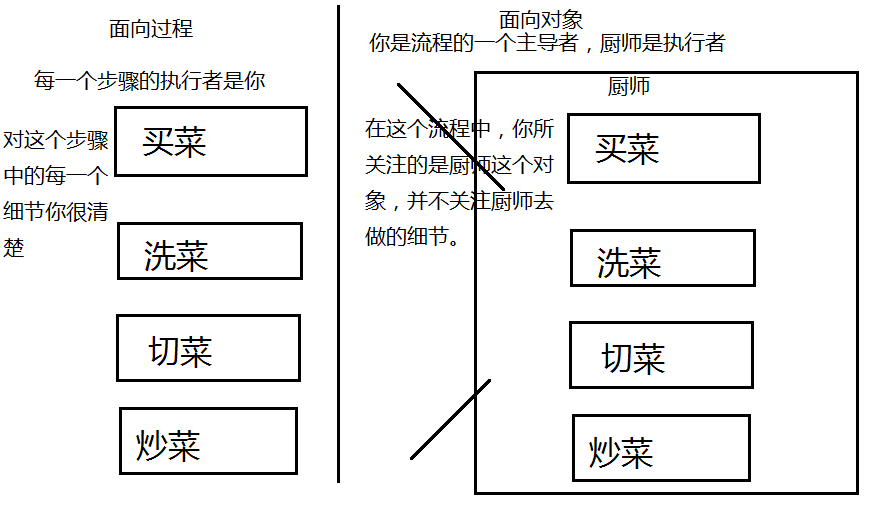
## 面向对象

### 面向对象和面向过程的比较

面向对象本身是相对于面向过程而言的。

面向过程和面向对象是思维方式。

面向过程注重的每一个步骤；面向对象注重的对象，只要获取到了这个对象，获取到这个对象所拥有的一切的功能。

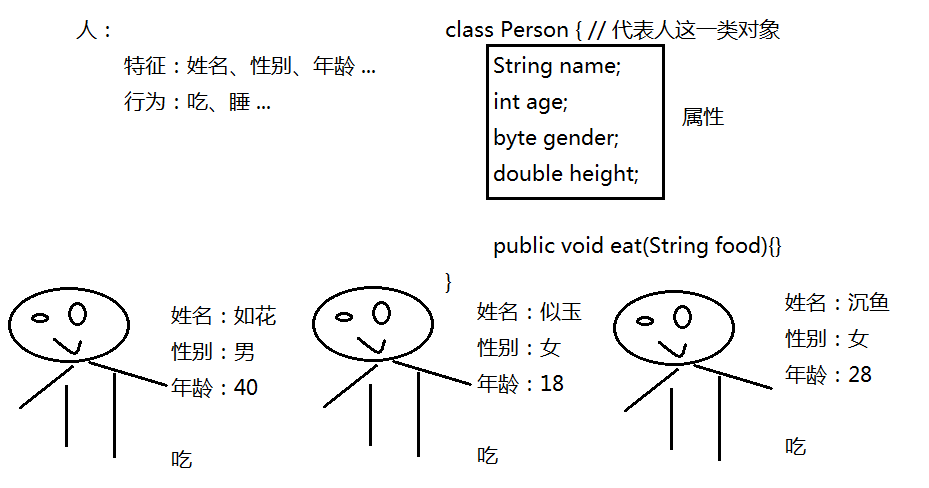


面向对象一定优于面向过程吗？---不一定 --- 如果处理的事务比较简单，适合于面向过程；如果处理的事务比较复杂，适合于用面向对象

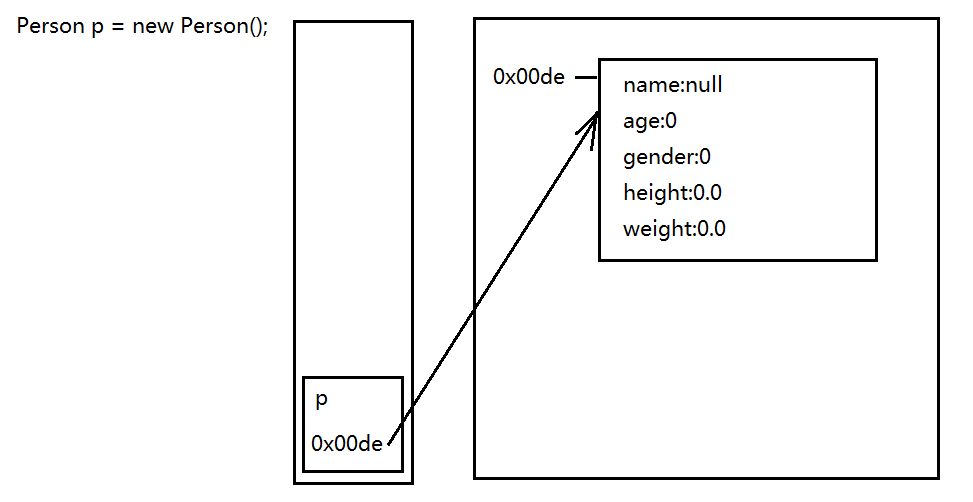
面向对象本身是基于面向过程的。

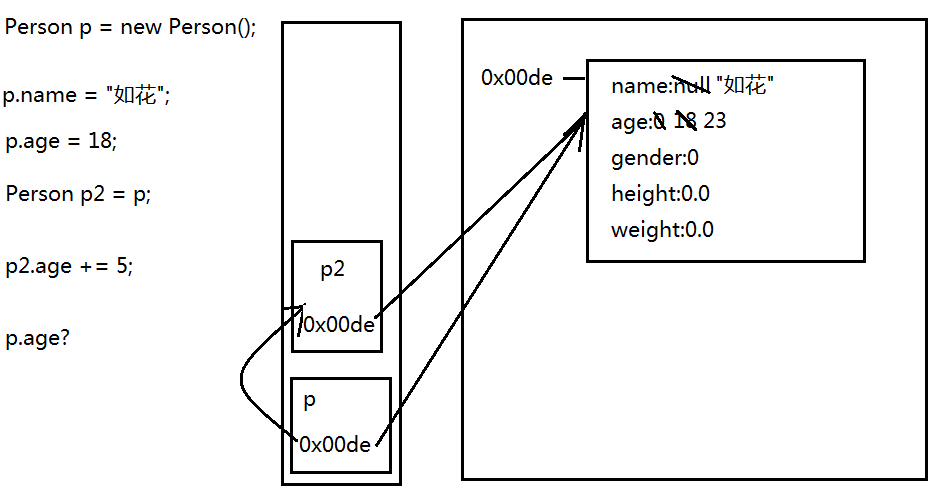
### 类与对象的关系

对一类对象进行概括，将概括的形式写成了类。--将对象的特征总结成了属性，将对象的行为总结了方法。



### 对象在内存中的存储





### 成员变量和局部变量的区别

1. 定义位置：成员变量定义在类内，局部变量定义在方法内

2. 作用范围：成员变量可以作用在整个类内，局部变量可以作用在方法内

3. 内存位置：成员变量是存储在堆内存中，并且在堆内存中会自动的赋予一个默认值；局部变量存储在栈内存中，不会自动给予默认值

4. 生命周期：成员变量在对象被创建的时候放到了堆内存中，对象被销毁的时候才被释放；局部变量在方法被调用的时候被创建，方法使用完成之后立即释放

### 构造方法

用于创建对象的方法 --- 与类同名而没有返回值类型

如果程序没有手动指定构造方法，那么默认添加一个无参构造方法

构造方法可以重载