复习：

1. 方法的重载：在同一个类中，方法名一样而参数列表不同 --- 参数个数或者对应位置上的参数类型不一样。

2. 方法的递归：在方法内部调用方法自己本身。--- 整体思想是逆推。 --- 实际上找这一项和之前一项或者是几项的关系

练习：

斐波那契数列：0,1,1,2,3,5,8,13... --- 输入一个数字n，获取第n项 --- 从第三项开始，每一项都是它之前两项的和

3. 方法的传参：如果传参传递的是基本类型，传递的实际的值；如果传参传递的是引用类型，传递的是地址。

4. 面向对象和面向过程的比较：如果一个流程中的每一步你都熟悉，这就是面向过程的（所有的事情都是自己做）；一件事儿让别人去做的，这就是面向对象。面向对象是基于面向过程的。面向对象和面向过程的使用场景不一样---如果是简单的事务使用面向过程；如果是复杂的事务使用面向对象

5. 类和对象的关系：类是对象的概括，对象是类的具体展现

6. 成员变量和局部变量的不同：定义位置、作用范围、内存位置、生命周期

7. 构造方法：用于创建指定类的对象。与类同名，没有返回值类型。 --- 如果一个类中没有手动指定构造方法，在编译的时候会默认添加一个无参构造。如果手动添加了构造方法，则不会在默认添加了。如果一个类只提供了含参构造，那么在创建对象的时候就只能使用含参构造。

### this关键字

this代替一个在活动的本类对象，可以把它认为是一个虚拟对象。

this用在这个类中，调用本类中的属性和方法的。

this语句 --- 用于在本类中调用本类其他的对应形式的构造方法 --- 必须放在构造方法的首行

练习：

定义一个类表示复数。（a + bi） --- 在这个类中定义2个属性，一个表示实部，另一个表示虚部 --- 提供对应的构造方法， 提供一个方法允许当前的复数和另一个复数求和

(a + bi) + (c + di) = (a + c) + (b + d)i

### 构造代码块

如果每一个构造方法中都有一些行为需要完成，那么这个时候可以将这些行为提取到构造代码块中。---无论使用哪一个构造方法，都会先于构造方法执行一次

### 局部代码块

在方法中，用{}括起来的代码。--- 限制变量的使用范围，缩短了变量的生命周期，以提高栈内存的利用率

## 面向对象的特征

封装 继承 多态 (抽象)

### 封装

体现形式：方法、内部类、属性的私有化 --- 将属性用private修饰，然后提供对外的get和set方法来获取或者设置属性，在方法中对属性进行限制使数据更加的符合场景要求

优势：提高代码的复用性。保证数据的合法性

age --- Person 0~130 Student >=3 Driver 18~65

name --- 5个汉字

### 权限修饰符

用于限制属性或者方法的使用范围的

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 本类中 | 子类中 | 同包类中 | 其他类中 |
| public | 可以 | 可以 | 可以 | 可以 |
| protected | 可以 | 可以 | 可以 | 不可以 |
| 默认 | 可以 | 同包子类可以 | 可以 | 不可以 |
| private | 可以 | 不可以 | 不可以 | 不可以 |