

【济南中心】JAVA 编程阶梯：基础篇之第六章

- **面向过程：**

面向过程开发，其实就是面向着具体的每一个步骤和过程，把每一个步骤和过程完成，然后由这些功能方法相互调用，完成需求。面向过程的代表语言：

c 语言

- **面向对象思想概述**

面向对象是基于面向过程的编程思想

- **面向对象思想特点**

- 1、是一种更符合我们思想习惯的思想
- 2、可以将复杂的事情简单化
- 3、将我们从执行者变成了指挥者（角色发生了转换）

下面我们根据面向和面向过程的概念和特点各举个例子：

面向过程：

- 1、打开手机
- 2、拨打电话
- 3、挂断电话

强调的是“打开手机”，“拨打电话”、“挂断电话”过程功能行为的重要性、

面向对象：

- 1、手机打开了
- 2、手机拨打电话
- 3、手机关闭了

强调手机是一个对象一个整体，具有打开，拨打号码，关闭的功能。

- **面向对象开发**

就是不断的创建对象，使用对象，指挥对象做事情。

- **面向对象设计**

其实就是在管理和维护对象之间的关系。

- **面向对象特征**

封装(encapsulation)

继承(inheritance)

多态(polymorphism)

- **类的定义**

生活中描述事物无非就是描述事物的属性和行为

属性 人的身高，体重等

行为 人可以学习，吃饭等

那我们定义类其实就是定义类的成员(成员变量和成员方法)

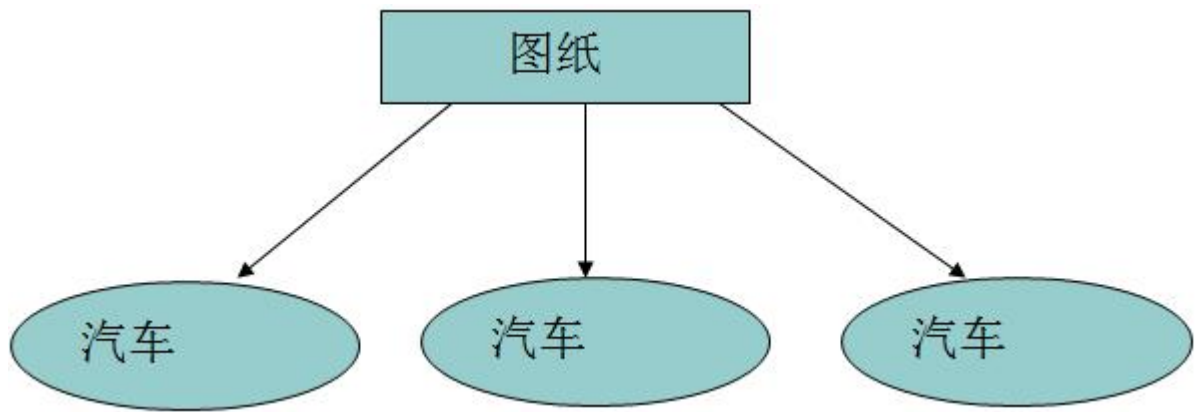
成员变量 就是事物的属性、

成员方法 就是事物的行为

- **类与对象关系**

类：是一组相关的属性和行为的集合，可以理解为构造对象的一个蓝图或者模版，是抽象的概念，比如：森林

对象：是该类事物的具体体现，是以类为模型创建的具体实例，是对类的一种具体化。比如：白桦树、松柏、梧桐



类是图纸，汽车就是堆内存中的对象。

使用：

A、创建对象并使用 格式：类名 对象名 = new 类名();

B、使用成员变量：对象名.变量名

C、使用成员方法：对象名.方法名()

给大家举个小栗子：

```
class Demo {[/align]
    int sum;

    public int getSum(int a, int b) {
        sum = a + b;
        return sum;
    }
}

class Test {
    public static void main(String[] args) {
        Demo d = new Demo();
        System.out.println(d.getSum(20, 20));
        System.out.println(d.sum);
    }
}
```

• 成员变量和局部变量的区别

A、在类中的位置不同

成员变量 类中方法外

局部变量 方法内或者方法声明上

B、内存中的位置不同

成员变量 堆内存

局部变量 栈内存

C、生命周期不同

成员变量 随着对象的存在而存在，随着对象的消失而消失

局部变量 随着方法的调用而存在，随着方法的调用完毕而消失

D、初始化值不同

成员变量 有默认的初始化值

局部变量没有默认的初始化值，必须先定义，赋值，才能使用

如果局部变量和成员变量的名称一致才会有就近原则，先找小范围，再找大范围。

基本数据类型变量包

括:byte,short,int,long,float,double,boolean,char

引用数据类型变量包括:数组,类,接口,枚举

• **方法的形式参数**

一般情况下基本类型和引用类型都可以作为形式参数，基本类型就是值传递

```
class Student {  
    public void show() {  
        System.out.println("show");  
    }  
}
```

```
class StudentDemo {  
    // 如果参数是一个类名，那么实际需要的是一个具体的对象
```

```

        public void method(Student s) {
            s.show();
        }
    }

    public class StudentTest {
        public static void main(String[] args) {
            StudentDemo sd = new StudentDemo();
            Student s = new Student();
            sd.method(s);

            // 多个匿名对象的写法
            new StudentDemo().method(s);
        }
    }

```

例子中提到了匿名对象，那么匿名对象又是什么？

简单来说匿名对象就是没有名字的对象是对象的一种简化表示形式

匿名对象的两种使用情况：

- 1、对象调用方法仅仅一次的时候
- 2、作为实际参数传递

• 封装的概述

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。

• 封装好处

- A、隐藏实现细节，提供公共的访问方式
- B、提高了代码的复用性
- C、提高安全性。

• 封装原则

- A、将不需要对外提供的内容都隐藏起来。
- B、把属性隐藏，提供公共方法对其访问。

• private 关键字的特点

- A、是一个权限修饰符
- B、可以修饰成员变量和成员方法
- C、被其修饰的成员只能在本类中被访问

- **封装和 private 的应用：**

- A、把成员变量用 private 修饰
- B、提供对应的 getXxx()和 setXxx()方法

private 仅仅是封装的一种体现形式,不能说封装就是私有

举例：

```
class Student {  
    private String name;  
    private int age;  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public int getAge() {  
        return age;  
    }  
  
    public void setAge(int age) {  
        this.age = age;  
    }  
}  
  
public class StudentTest {  
    public static void main(String[] args) {  
        Student stu = new Student();  
        // stu.name="小王" //因为成员属性被私有化了，我  
        们不能直接访问只能通过 set() 方法进行设置  
        stu.setName("小王");  
    }  
}
```

```
        stu.setAge(18);
        System.out.println("姓名:" + stu.getName() + " 年
龄:" + stu.getAge());
    }
}
```

- **this 关键字**

代表所在类的对象引用，哪个对象调用了 this 所在的函数，this 就代表哪个对象

- **this 的应用场景**

用来区分成员变量和局部变量重名，这个标识可以省略不写。但是，当局部变量和成员变量同名时，必须用 this.来标识成员变量



识别二维码 关注黑马程序员视频库
免费获得更多 IT 资源