

1.3 什么是类

类是对现实生活中一类具有**共同属性**和**行为**的事物的抽象

类的特点：

- 类是对象的数据类型
- 类是具有相同属性和行为的一组对象的集合

1.7 类的定义

类的重要性：是Java程序的基本组成单位

类是什么：是对现实生活中一类具有**共同属性**和**行为**的事物的抽象，确定对象将会拥有的属性和行为

类的组成：**属性**和**行为**

- 属性：在类中通过**成员变量**来体现（类中方法外的变量）
- 行为：在类中通过**成员方法**来体现（和前面的方法相比去掉static关键字即可）

3.2 成员变量和局部变量区别

区别	成员变量	局部变量
类中位置不同	类中方法外	方法内或者方法声明上
内存中位置不同	堆内存	栈内存
生命周期不同	随着对象的存在而存在，随着对象的消失而消失	随着方法的调用而存在，随着方法的调用完毕而消失
初始化值不同	有默认的初始化值	没有默认的初始化值，必须先定义，赋值，才能使用

方法声明是指方法形参

4.1 private关键字

- 是一个权限修饰符
- 可以修饰成员（成员变量和成员方法）
- 作用是保护成员不被别的类使用，被**private**修饰的成员只在本类中才能访问

针对**private**修饰的成员变量，如果需要被其他类使用，提供相应的操作

- 提供“**get变量名()**”方法，用于获取成员变量的值，方法用**public**修饰
- 提供“**set变量名(参数)**”方法，用于设置成员变量的值，方法用**public**修饰

4.3 this关键字

- ① this修饰的变量用于指代成员变量
 - 方法的形参如果与成员变量同名，不带this修饰的变量指的是形参，而不是成员变量
 - 方法的形参没有与成员变量同名，不带this修饰的变量指的是成员变量
- ② 什么时候使用this呢？**解决局部变量隐藏成员变量**
- ③ this：代表所在类的对象引用
 - 记住：方法被哪个对象调用，this就代表哪个对象

```
public class Student {  
    private String name;  
  
    public String getName() {  
        return name;  
    }  
    public void setName(String name) {  
        this.name = name;  
    }  
}  
  
public class StudentDemo {  
    public static void main(String[] args) {  
        Student s1 = new Student();  
        s1.setName("林青霞");  
        Student s2 = new Student();  
        s2.setName("张曼玉");  
    }  
}
```

4.5 封装

1. 封装概述

是面向对象三大特征之一（**封装，继承，多态**）

是面向对象编程语言对客观世界的模拟，客观世界里成员变量都是隐藏在对象内部的，外界是无法直接操作的

2. 封装原则

将类的某些信息隐藏在类内部，不允许外部程序直接访问，而是通过该类提供的方法来实现对隐藏信息的操作和访问
成员变量**private**，提供对应的**getXxx()/setXxx()**方法

3. 封装好处

通过方法来控制成员变量的操作，提高了代码的安全性

把代码用方法进行封装，提高了代码的复用性

5.2 构造方法的注意事项

- ① 构造方法的创建
 - 如果没有定义构造方法，系统将给出一个**默认的无参数构造方法**
 - 如果定义了构造方法，系统将不再提供默认的构造方法
- ② 构造方法的重载
 - 如果自定义了带参构造方法，还要使用无参数构造方法，就必须再写一个无参数构造方法
- ③ 推荐的使用方式
 - **无论是否使用，都手工书写无参数构造方法**