

# 第七章 声明和使用构造器

## 1. 构造器的概述

1. 构造器：也叫构造方法，特殊的方法
2. 普通方法VS构造器方法

代码块

```
1  1. 普通方法
2      [修饰符] 返回类型  方法名称([形参类别]){ }
3  2. 构造方法
4      [修饰符]    类名称([形参类别]){ }
```

### 3. 构造方法的特点

- a. 无返回类型
- b. 方法名称和类名称一样
- c. 调用方式不一样
  - i. new 类名称([实参类别])：常用
  - ii. this([实参列表])：不常用

### 4. 构造器的注意事项（会背）

- a. 一个类如果没有任何构造器，编译时会添加默认构造器
- b. 如果一个类有了构造器，编译时不会添加默认构造器

### 5. 构造的作用

- a. 通过构造器可以创建对象
- b. 通过构造器可以为属性批量赋值

## 2. 默认构造器

代码块

```
1  package com.powernode.constructor14;
2
3  class Teacher{
4      /**
5          * 1. 一个类如果没有任何构造器，编译时会添加默认构造器
```

```

6      *   2. 如果一个类有了构造器，编译时不会添加默认构造器
7      *   3.构造器语法：[修饰符] 类名称([形参类别]){}
8      *   4.javap Teacher(命令查看.class文件)
9      *   5.注意：默认构造器的修饰符和类的修饰符一样
10     */
11     Teacher(){
12         System.out.println("Teacher.Teacher");
13     }
14
15 }
16 public class Test {
17     public static void main(String[] args) {
18         //类名称 对象名称 = new 类名称([实参列表]);
19         Teacher teacher = new Teacher();
20         System.out.println(teacher);
21     }
22 }

```

### 3. 通过构造器为属性批量赋值

代码块

```

1  package com.powernode.constructor15;
2
3  class Teacher{
4      private String name;
5      private int age;
6
7      Teacher(String name,int age){
8          this.name = name;
9          this.age = age;
10     }
11
12     /* public void setName(String name) {
13         this.name = name;
14     }
15
16     public void setAge(int age) {
17         this.age = age;
18     }*/
19 }
20 public class Test {
21     public static void main(String[] args) {
22         //创建对象的时候为属性批量赋值
23         Teacher teacher = new Teacher("zs",23);
24

```

```
25     }  
26 }
```

## 4. 一个实体类通常的写法

代码块

```
1  package com.powernode.constructor16;  
2  class Teacher{  
3      //1.属性  
4      private String name;  
5      private int age;  
6      private char sex;  
7      //2.构造方法  
8      Teacher(){  
9      }  
10  
11     Teacher(String name,int age,char sex){  
12         this.name = name;  
13         this.age = age;  
14         this.sex = sex;  
15     }  
16     //3.set和get方法  
17     public void setName(String name) {  
18         this.name = name;  
19     }  
20  
21     public String getName() {  
22         return name;  
23     }  
24  
25     public void setAge(int age) {  
26         this.age = age;  
27     }  
28  
29     public int getAge() {  
30         return age;  
31     }  
32  
33     public void setSex(char sex) {  
34         this.sex = sex;  
35     }  
36  
37     public char getSex() {  
38         return sex;  
39     }  
}
```

```

40     //4. 提供自我描述的信息
41     public String getDetails(){
42         return "姓名: " + name + "\t年龄: " + age + "\t性别: " + sex;
43     }
44 }
45 public class Test {
46     public static void main(String[] args) {
47         //不知道属性值: 无参构造器
48         Teacher t1 = new Teacher();
49         //知道属性值: 全参构造器
50         Teacher t2 = new Teacher("zs",23,'女');
51         //修改name的值
52         t2.setName("ww");
53         System.out.println(t2.getName());
54
55     }
56 }

```

## 5. 构造器重载

代码块

```

1  package com.powernode.constructor17;
2  class Teacher{
3      private String name;
4      private int age;
5      private char sex;
6      /**
7       * - 方法重载
8       *     1. 在同一个类中
9       *     2. 方法名称相同
10      *     3. 形参类别不同
11      *         1. 个数不同
12      *         2. 类型不同
13      *         3. 顺序不同
14      */
15     Teacher(){
16     }
17
18     Teacher(String name ){
19         this.name = name;
20     }
21
22     Teacher(String name ,int age){

```

```

23         this.name = name;
24         this.age = age;
25     }
26
27     Teacher(String name ,int age,char sex){
28         this.name = name;
29         this.age = age;
30         this.sex = sex;
31     }
32
33     public String getDetails(){
34         return "姓名: " + name + "\t年龄: " + age + "\t性别: " + sex;
35     }
36
37 }
38 public class Test {
39     public static void main(String[] args) {
40         Teacher t1 = new Teacher();
41         System.out.println(t1.getDetails());
42
43         Teacher t2 = new Teacher("zs");
44         System.out.println(t2.getDetails());
45
46         Teacher t3 = new Teacher("zs", 23);
47         System.out.println(t3.getDetails());
48
49         Teacher t4 = new Teacher("zs", 23, '女');
50         System.out.println(t4.getDetails());
51
52     }
53 }

```

## 6. this调用重载构造器（使用不多）

代码块

```

1  package com.powernode.constructor18;
2  class Teacher{
3      private String name;
4      private int age;
5      private char sex;
6      /**
7       * - 方法重载
8       *     1.在同一个类中
9       *     2.方法名称相同
10      *     3.形参类别不同

```

```
11      *      1.个数不同
12      *      2.类型不同
13      *      3.顺序不同
14      */
15      Teacher(){
16      }
17      /*public void m1(){
18          m2(2,3);
19      }
20      public void m2(int x,int y){}*/
21      Teacher(String name ){
22          //this.name = name;
23          //Teacher(name,0);
24          this(name,0);
25      }
26
27      Teacher(String name ,int age){
28          /*this.name = name;
29          this.age = age;*/
30          this(name,age,'\u0000');
31      }
32
33      Teacher(String name ,int age,char sex){
34          this.name = name;
35          this.age = age;
36          this.sex = sex;
37      }
38
39      public String getDetails(){
40          return "姓名: " + name + "\t年龄: " + age + "\t性别: " + sex;
41      }
42
43  }
44  public class Test {
45      public static void main(String[] args) {
46          Teacher t1 = new Teacher();
47          System.out.println(t1.getDetails());
48
49          Teacher t2 = new Teacher("zs");
50          System.out.println(t2.getDetails());
51
52          Teacher t3 = new Teacher("zs", 23);
53          System.out.println(t3.getDetails());
54
55          Teacher t4 = new Teacher("zs", 23, '女');
56          System.out.println(t4.getDetails());
57      }
```

```
58     }
59 }
```

## 7. this调用重载构造器，必须放在第一行

代码块

```
1  package com.powernode.constructor19;
2  class Teacher{
3      private String name;
4      private int age;
5      private char sex;
6
7      Teacher(){
8          /**
9           * Call to 'this()' must be first statement in constructor body
10          * 调用'this()'必须是构造函数主体中的第一条语句
11          */
12          //System.out.println("-----");
13          this("zs", 23, '男');
14      }
15
16
17      Teacher(String name ,int age,char sex){
18          this.name = name;
19          this.age = age;
20          this.sex = sex;
21      }
22
23      public String getDetails(){
24          return "姓名: " + name + "\t年龄: " + age + "\t性别: " + sex;
25      }
26
27  }
28  public class Test {
29      public static void main(String[] args) {
30          Teacher t1 = new Teacher();
31          System.out.println(t1.getDetails());
32
33
34
35          Teacher t4 = new Teacher("zs", 23, '女');
36          System.out.println(t4.getDetails());
37
38      }
39  }
```

# 作业

## 1. 练习题：

- a. 声明一个Student类，将name、gender、age、id、score属性均为私有，并赋值，say方法设为公有。
- b. 在另一个TestStudent类中的main方法中，创建Student对象，并试着访问对象中的属性。
- c. 对所有属性提供公有的get/set方法，在TestStudent类的main方法中，访问调用set方法为属性设值，再调用get方法将值获取并打印输出。

## 2. 练习题

- a. 声明Cat类，将name,age,weight属性均为私有，并赋值，并提供公有的get/set方法，同时将say方法设为公有。
- b. 在另一个TestCat类中的main方法中，创建Cat对象，调用set方法为属性设值，再调用get方法将值获取并打印输出。

## 3. 无参构造器

- a. 修改Student类，添加无参数的构造器，用来初始所有属性
- b. 创建Student对象，调用say方法打印输出结果。

## 4. 有参构造器

- a. 修改Student类，添加带参数的构造器，用参数来初始所有属性。
- b. 创建Student对象，调用say方法打印输出结果。

## 5. 构造器重载

- a. 修改Student类，添加一到多个带参数的重载构造器，并使用构造器来初始所有属性。
- b. 创建Student对象,调用重载构造器，调用say方法打印输出结果。

## 6. this调用构造器

- a. 修改Student类，在构造器中使用this调用其他重载构造器，以完全初始所有属性。
- b. 用不同数量的参数分别创建多个不同的Student对象，并分别调用say方法打印输出结果。

## 7. 综合练习

- a. Room类
  - i. 属性：房号（rNO）、面积（area）、地址（address）
  - ii. 使用封装



- iii. 构造器：定义4个重载构造器
- iv. 使用this（）调用重载构造器
- v. 方法：getDetails 返回String，描述房间详细信息

b. TestRoom类

分别用4个构造器创建4个room对象，打印对象的详细信息

8. 综合练习

a. 编写一个Computer类，包括：

- i. 属性cpu、内存（memory）和硬盘（hardDisk）
- ii. 带参数的构造器用来初始化相关属性
- iii. 为各属性提供对应的get/set方法
- iv. 提供getDetails方法，用来返回自我描述的字符串

b. 编写Test类，在main 方法中分别创建两个Computer对象，并分别调用对象上的getDetails方法输出自我描述。