

Java基础8 构造函数

构造函数

Java构造函数，也叫构造方法，是java中一种特殊的函数。函数名与相同，无返回值。

作用

一般用来初始化成员属性和成员方法的，即new对象产生后，就调用了对象了属性和方法。

特点

- 构造函数是对象一建立就运行，给对象初始化，就包括属性，执行方法中的语句。
- 而一般函数是对象调用才执行，用".方法名"的方式，给对象添加功能。
- 一个对象建立，构造函数只运行一次。
- 而一般函数可以被该对象调用多次。

注意事项

- 函数名与类名相同
- 不用定义返回值类型。（不同于void类型返回值，void是没有具体返回值类型；构造函数是连类型都没有）
- 不可以写return语句。（返回值类型都没有，也就不需要return语句了）
- 一般函数不能调用构造函数，只有构造函数才能调用构造函数。

示例

无参构造函数中只定义了一个方法。new对象时，就调用与之对应的构造函数，执行这个方法。不必写“方法名”。

```
package javastudy;

public class ConfunDemo {
    public static void main(String[] args) {
        //输出Hello world。new对象一建立，就会调用对应的构造函数Confun()，并执行其中的println语句。
        Confun c1=new Confun();
    }
}
class Confun{
    Confun(){
        //定义构造函数，输出Hello world
        System.out.println("Hellow world");
    }
}
```

输出：

```
Hello World
```

有参构造函数，在new对象时，将实参值传给private变量，相当于完成setter功能。

```
package javastudy;

public class ConfunDemo3 {
    public static void main(String[] args){
        //实例化对象时，new Person()里直接调用Person构造函数并转转实参，相当于setter功能
        Person z = new Person("aerchi",18);
        z.show();
    }
}

class Person{
    private String name;
    private int age;
    //有参数构造函数，实现给private成员变量传参数值的功能
    public Person(String n,int m){
        name=n;
        age=m;
    }
    //getter
    //实例化对象时，完成了set功能后，需要getter，获取实参值。
    public String getName(){
        return name;
    }
    public int getAget(){
        return age;
    }

    //获取private值后，并打印输出
    public void show(){
        System.out.println(name+"\n"+age);
    }
}
```

输出：

```
aerchi
18
```

一个对象建立后，构造函数只运行一次。如果想给对象的值再赋新的值，就要使用set和get方法，此时是当做一般函数使用

```
package javastudy;

public class ConfunDemo4 {
    public static void main(String[] args) {
        PersonDemo s=new PersonDemo("张三",18); //new对象时，即调用对应的构造函数，并传值。同时，不能new同一个对象多次，否则会报错。
        s.setName("李四"); //对象建立后，想变更值时，就要用set/get方法，重新设置新的值
        s.setName("王二麻子"); //并可调用对象多次。
    }
}
```

```

        s.print();
    }
}
class PersonDemo{
    private String name;
    private int age;
    PersonDemo(String n,int m){          //建立有参构造函数，用于给两个private变量name、
age赋值，同时输出值
        name=n;
        age=m;
        System.out.println("姓名: "+name+"\n"+"年龄: "+age);
    }
    public void setName(String x){      //set方法，用于再次给name赋值
        name=x;
    }
    public String getName(){           //get方法，用于获取name的赋值
        return name;
    }
    public void print(){
        System.out.println(name);
    }
}

```

输出：

```

姓名：张三
年龄：18
王二麻子

```

默认构造器

当一个类中没有定义构造函数时，系统会给该类中加一个默认的空参数的构造函数，方便该类初始化。只是该空构造函数是隐藏不见的。

如下，Person(){}这个默认构造函数是隐藏不显示的。

```

class Person
{
    //Person(){}
}

```

当在该类中自定义了构造函数，默认构造函数就没有了。如果仍要构造函数，则需要自己在类中手动添加。

构造函数的重载

构造函数也是函数的一种，同样具备函数的重载（Overloading）特性。

```

class Person
{
    private String name;
    private int age;
}

```

```

    Person()
    {
        System.out.println("A:name="+name+"::age="+age);
    }

    Person(String n)
    {
        name = n;
        System.out.println("B:name="+name+"::age="+age);
    }

    Person(String n,int a)
    {
        name=n;
        age=a;
        System.out.println("C:name="+name+"::age="+age);
    }
}

class PersonDemo2
{
    public static void main(String[] args)
    {
        Person p1=new Person();
        Person p2=new Person("aerchi");
        Person p3=new Person("aerchi",18);
    }
}

```

输出结果:

```

A:name=null::age=0
B:name=aerchi::age=0
C:name=aerchi::age=18

```

构造函数的使用

子类中无参构造函数继承父类中无参构造函数时，父类参数是private的，无法直接访问。需要在父类中使用get方法来调用私有变量值。

```

package javastudy;

public class ConfunDemo5 {

    public static void main(String[] args) {
        Pupil z=new Pupil();
        z.show();
    }
}

class Student{                                //父类Student
    private String name;
    private int height;
    public Student()

```

```

    {
        this.name="";
        this.height=0;
    }
    public String getName(){
        return name;
    }
    public int getHeight(){
        return height;
    }
}
class Pupil extends Student{    //子类Pupil
    private int score;
    public Pupil(){                //无参构造函数Pupil()直接继承了父类中的无参构造函数Student(), 但是父类中的name、height是private的
        score=0;
    }
    public void show(){
        System.out.print("姓名: "+getName()+"\n身高: "+getHeight()+"\n分数: "+score);    //输出时, 直接用get方法名。
    }
}

```

使用super调用父类的构造函数

```

package javastudy;

public class ConfunDemo5 {

    public static void main(String[] args) {
        Pupil z=new Pupil("王二麻子",100,200);
        z.show();

        Pupil w=new Pupil();
        w.show();
    }
}
class Student{                //父类Student
    public String name;
    public int height;
    public Student()
    {
        this.name="";
        this.height=0;
    }
    public Student(String n,int m)
    {
        name=n;
        height=m;
    }
}
class Pupil extends Student{    //子类Pupil
    private int score;
    public Pupil(){
        super("刘德华",501);    //使用super调用父类Student(String n,int m)方法, 同时传递实际数值。super必须写在方法的首行。如果这里写super(), 则调用的是父类中的Student()方法。
        score=0;
    }
}

```

```
}  
public Pupil(String x,int y,int z){           //  
    super(x,y);           //使用super调用父类Student(String n,int m)方法,其  
    中super中的参数名称必须与构造函数中的参数名称一致。  
    score=z;  
}  
public void show(){  
    System.out.println("姓名: "+name+"\n身高: "+height+"\n分数: "+score);  
}  
}
```

输出:

```
姓名: 王二麻子  
身高: 100  
分数: 200  
姓名: 刘德花  
身高: 501  
分数: 0
```