

java 中创建对象的方法

有 4 种显式地创建对象的方式：

- 1.用 **new 语句** 创建对象，这是最常用的创建对象的方式。
- 2.运用**反射手段**，调用 `java.lang.Class` 或者 `java.lang.reflect.Constructor` 类的 **`newInstance()`** 实例方法。
- 3.调用对象的 **`clone()` 方法**。
- 4.运用**反序列化手段**，调用 `java.io.ObjectInputStream` 对象的 `readObject()` 方法。

下面演示了用前面 3 种方式创建对象的过程。

```
1. public class Customer implements Cloneable{
2.     private String name;
3.     private int age;
4.     public Customer(){
5.         this("unknown",0);
6.         System.out.println("call default constructor");
7.     }
8.     public Customer(String name,int age){
9.         this.name=name;
10.        this.age=age;
11.        System.out.println("call second constructor");
12.    }
13.    public Object clone()throws CloneNotSupportedException{
14.        return super.clone();
15.    }
16.    public boolean equals(Object o){
17.        if(this==o)return true;
18.        if(! (o instanceof Customer)) return false;
19.        final Customer other=(Customer)o;
20.        if(this.name.equals(other.name) && this.age==other.age)
21.            return true;
22.        else
23.            return false;
24.    }
25.    public String toString(){
26.        return "name="+name+",age="+age;
27.    }
28.    public static void main(String args[])throws Exception{
```

```
29. //运用反射手段创建 Customer 对象
30. Class objClass=Class.forName("Customer");
31. Customer c1=(Customer)objClass.newInstance(); //会调用 Customer 类的默认构造方法
32. System.out.println("c1: "+c1); //打印 name=unknown,age=0
33.
34. //用 new 语句创建 Customer 对象
35. Customer c2=new Customer("Tom",20);
36. System.out.println("c2: "+c2); //打印 name=tom,age=20
37.
38. //运用克隆手段创建 Customer 对象
39. Customer c3=(Customer)c2.clone(); //不会调用 Customer 类的构造方法
40. System.out.println("c2==c3 : "+(c2==c3)); //打印 false
41. System.out.println("c2.equals(c3) : "+c2.equals(c3)); //打印 true
42. System.out.println("c3: "+c3); //打印 name=tom,age=20
43. }
44. }
```

以上程序的打印结果如下：

```
call second constructor

call default constructor

c1: name=unknown,age=0

call second constructor

c2: name=Tom,age=20

c2==c3 : false

c2.equals(c3) : true

c3: name=Tom,age=20
```

从以上打印结果看出，用 **new** 语句或 **Class** 对象的 **newInstance()** 方法创建 **Customer** 对象时，都会执行 **Customer** 类的构造方法，而用对象的 **clone()** 方法创建 **Customer** 对象时，不会执行 **Customer** 类的构造方法。（区别）

除了以上 4 种显式地创建对象的方式以外，在程序中还可以隐含地创建对象，包括以下几种情况：

1.对于 java 命令中的每个命令行参数，Java 虚拟机都会创建相应的 **String** 对象，并把它们组织到一个 **String** 数组中，再把该数组作为参数传给程序入口 `main(String args[])` 方法。

2.程序代码中的 **String** 类型的直接数对应一个 **String** 对象，例如：

```
1. String s1="Hello";
2. String s2="Hello"; //s2 和 s1 引用同一个 String 对象
3. String s3=new String("Hello");
4. System.out.println(s1==s2); //打印 true
5. System.out.println(s1==s3); //打印 false
```

执行完以上程序，内存中实际上只有两个 **String** 对象，一个是直接数，由 **Java** 虚拟机隐含地创建，还有一个通过 `new` 语句显式地创建。

3.字符串操作符 “+” 的运算结果为一个新的 **String** 对象。例如：

```
1. String s1="H";
2. String s2="ello";
3. String s3=s1+s2; //s3 引用一个新的 String 对象
4. System.out.println(s3=="Hello"); //打印 false
5. System.out.println(s3.equals("Hello")); //打印 true
```

4.当 **Java** 虚拟机加载一个类时，会隐含地创建描述这个类的 **Class** 实例。