**构造方法**

当一个对象被创建时候，构造方法用来初始化该对象。构造方法和它所在类的名字相同，但构造方法没有返回值。

通常会使用构造方法给一个类的实例变量赋初值，或者执行其它必要的步骤来创建一个完整的对象。

不管你是否自定义构造方法，所有的类都有构造方法，因为 Java 自动提供了一个默认构造方法，默认构造方法的访问修饰符和类的访问修饰符相同(类为 public，构造函数也为 public；类改为 protected，构造函数也改为 protected)。

一旦你定义了自己的构造方法，默认构造方法就会失效。

实例

下面是一个使用构造方法的例子：

// 一个简单的构造函数

class MyClass {

int x;

// 以下是构造函数

MyClass() {

x = 10;

}

}

你可以像下面这样调用构造方法来初始化一个对象：

ConsDemo.java 文件代码：

public class ConsDemo {

public static void main(String[] args) {

MyClass t1 = new MyClass();

MyClass t2 = new MyClass();

System.out.println(t1.x + " " + t2.x);

}

}

大多时候需要一个有参数的构造方法。

实例

下面是一个使用构造方法的例子：

// 一个简单的构造函数

class MyClass {

int x;

// 以下是构造函数

MyClass(int i ) {

x = i;

}

}

你可以像下面这样调用构造方法来初始化一个对象：

ConsDemo.java 文件代码：

public class ConsDemo {

public static void main(String[] args) {

MyClass t1 = new MyClass( 10 );

MyClass t2 = new MyClass( 20 );

System.out.println(t1.x + " " + t2.x);

}

}

运行结果如下：

10 20