**成员变量（实例变量）**

成员变量声明在一个类中，但在方法、构造方法和语句块之外。

当一个对象被实例化之后，每个成员变量的值就跟着确定。

成员变量在对象创建的时候创建，在对象被销毁的时候销毁。

成员变量的值应该至少被一个方法、构造方法或者语句块引用，使得外部能够通过这些方式获取实例变量信息。

成员变量可以声明在使用前或者使用后。

访问修饰符可以修饰成员变量。

成员变量对于类中的方法、构造方法或者语句块是可见的。一般情况下应该把成员变量设为私有。通过使用访问修饰符可以使成员变量对子类可见。

成员变量具有默认值。数值型变量的默认值是0，布尔型变量的默认值是 false，引用类型变量的默认值是 null。变量的值可以在声明时指定，也可以在构造方法中指定；

成员变量可以直接通过变量名访问。但在静态方法以及其他类中，就应该使用完全限定名：ObjectReference.VariableName。

成员变量的声明语法为：

accessModifier type variableName;

accessModifier --表示访问修饰符，可以是 public、protected、private 或默认访问级别（即没有显式指定访问修饰符）。

type -- 表示变量的类型。

variableName -- 表示变量的名称。

与局部变量不同，成员变量的值在创建对象时被分配，即使未对其初始化，它们也会被赋予默认值，例如 int 类型的变量默认值为 0，boolean 类型的变量默认值为 false。

成员变量可以通过对象访问，也可以通过类名访问（如果它们是静态成员变量）。如果没有显式初始化成员变量，则它们将被赋予默认值。可以在构造函数或其他方法中初始化成员变量，或者通过对象或类名访问它们并设置它们的值。

实例

以下实例我们声明了两个成员变量 a 和 b，并对其进行了访问和设置。注意，我们可以通过对象访问成员变量，也可以通过类名访问静态成员变量。

实例

public class RunoobTest {

private int a; // 私有成员变量a

public String b = "Hello"; // 公有成员变量b

public static void main(String[] args) {

RunoobTest obj = new RunoobTest(); // 创建对象

obj.a = 10; // 访问成员变量a，并设置其值为10

System.out.println("a = " + obj.a);

obj.b = "World"; // 访问成员变量b，并设置其值为"World"

System.out.println("b = " + obj.b);

}

}

以上实例编译运行结果如下:

a = 10

b = World

以下实例我们声明了两个成员变量 name 和 salary，并对其进行了访问和设置。

Employee.java 文件代码：

import java.io.\*;

public class Employee{

// 这个成员变量对子类可见

public String name;

// 私有变量，仅在该类可见

private double salary;

//在构造器中对name赋值

public Employee (String empName){

name = empName;

}

//设定salary的值

public void setSalary(double empSal){

salary = empSal;

}

// 打印信息

public void printEmp(){

System.out.println("名字 : " + name );

System.out.println("薪水 : " + salary);

}

public static void main(String[] args){

Employee empOne = new Employee("RUNOOB");

empOne.setSalary(1000.0);

empOne.printEmp();

}

}

以上实例编译运行结果如下:

$ javac Employee.java

$ java Employee

名字 : RUNOOB

薪水 : 1000.0