**Java 变量类型**

在Java语言中，所有的变量在使用前必须声明。声明变量的基本格式如下：

type identifier [ = value][, identifier [= value] ...] ;

格式说明：type为Java数据类型。identifier是变量名。可以使用逗号隔开来声明多个同类型变量。

以下列出了一些变量的声明实例。注意有些包含了初始化过程。

int a, b, c;         // 声明三个int型整数：a、 b、c。

int d = 3, e, f = 5; // d声明三个整数并赋予初值。

byte z = 22;         // 声明并初始化z。

double pi = 3.14159; // 声明了pi。

char x = 'x';        // 变量x的值是字符'x'。

Java语言支持的变量类型有：

* 局部变量
* 成员变量
* 类变量

**Java局部变量**

* 局部变量声明在方法、构造方法或者语句块中；
* 局部变量在方法、构造方法、或者语句块被执行的时候创建，当它们执行完成后，变量将会被销毁；
* 访问修饰符不能用于局部变量；
* 局部变量只在声明它的方法、构造方法或者语句块中可见；
* 局部变量是在栈上分配的。
* 局部变量没有默认值，所以局部变量量被声明后，必须经过初始化，才可以使用。

**实例1**

在以下实例中age是一个局部变量。定义在pubAge()方法中，它的作用域就限制在这个方法中。

public class Test{

public void pupAge(){

int age = 0;

age = age + 7;

System.out.println("Puppy age is : " + age);

}

public static void main(String args[]){

Test test = new Test();

test.pupAge();

}

}

以上实例编译运行结果如下：

Puppy age is: 7

**实例2**

在下面的例子中age变量没有初始化，所以在编译时出错。

public class Test{

public void pupAge(){

int age;

age = age + 7;

System.out.println("Puppy age is : " + age);

}

public static void main(String args[]){

Test test = new Test();

test.pupAge();

}

}

以上实例编译运行结果如下:

Test.java:4:variable number might not have been initialized

age = age + 7;

^

1 error

**实例变量**

* 实例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，每个实例变量的值就跟着确定；
* 实例变量在对象创建的时候创建，在对象被销毁的时候销毁；
* 实例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；
* 实例变量可以声明在使用前或者使用后；
* 访问修饰符可以修饰实例变量；
* 实例变量对于类中的方法、构造方法或者语句块是可见的。一般情况下应该把实例变量设为私有。通过使用访问修饰符可以使实例变量对子类可见；
* 实例变量具有默认值。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；
* 实例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：ObejectReference.VariableName。

实例：

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("name  : " + name );

      System.out.println("salary :" + salary);

   }

   public static void main(String args[]){

      Employee empOne = new Employee("Ransika");

      empOne.setSalary(1000);

      empOne.printEmp();

   }

}

以上实例编译运行结果如下:

name : Ransika

salary :1000.0

**类变量（静态变量）**

* 类变量也称为静态变量，在类中以static关键字声明，但必须在方法构造方法和语句块之外。
* 无论一个类创建了多少个对象，类只拥有类变量的一份拷贝。
* 静态变量除了被声明为常量外很少使用。常量是指声明为publc/private，final和static类型的变量。常量初始化后不可改变。
* 静态变量储存在静态存储区。经常被声明为常量，很少单独使用static声明变量。
* 静态变量在程序开始时创建，在程序结束时销毁。
* 与实例变量具有相似的可见性。但为了对类的使用者可见，大多数静态变量声明为public类型。
* 默认值和实例变量相似。数值型变量默认值是0，布尔型默认值是false，引用类型默认值是null。变量的值可以在声明的时候指定，也可以在构造方法中指定。此外，静态变量还可以在静态语句块中初始化。
* 静态变量可以通过：*ClassName.VariableName*的方式访问。
* 类变量被声明为public static final类型时，类变量名称必须使用大写字母。如果静态变量不是public和final类型，其命名方式与实例变量以及局部变量的命名方式一致。

实例：

import java.io.\*;

public class Employee{

   //salary是静态的私有变量

   private static double salary;

   // DEPARTMENT是一个常量

   public static final String DEPARTMENT = "Development ";

   public static void main(String args[]){

      salary = 1000;

      System.out.println(DEPARTMENT+"average salary:"+salary);

   }

}

以上实例编译运行结果如下:

Development average salary:1000

**注意：**如果其他类想要访问该变量，可以这样访问：Employee.DEPARTMENT。

本章节中我们学习了Java的变量类型，下一章节中我们将介绍Java修饰符的使用。