**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。

局部变量必须在使用前声明，并且不能被访问修饰符修饰，因为它们的作用域已经被限制在了声明它们的方法、代码块或构造函数中。

局部变量只在声明它的方法、构造方法或者语句块中可见，不能被其他方法或代码块访问。

局部变量是在栈上分配的。

局部变量没有默认值，所以局部变量被声明后，必须经过初始化，才可以使用。

局部变量的声明语法为：

type variableName;

type -- 表示变量的类型。

variableName -- 表示变量的名称。

局部变量必须在使用之前进行初始化，否则编译器会报错。初始化可以在声明时或后面的代码中进行。如果在声明时未初始化，变量将被赋予默认值，如 int 类型的变量默认为 0，boolean 类型的变量默认为 false，引用类型的变量默认为 null。

局部变量只在声明它的方法、构造方法或语句块内可见，其他方法、构造方法或语句块不能访问该局部变量。当方法、构造方法或语句块执行完毕后，局部变量将被销毁，其占用的内存也会被释放。

实例

以下是一个简单的例子，展示了局部变量的使用：

实例

public class LocalVariablesExample {

public static void main(String[] args) {

int a = 10; // 局部变量a的声明和初始化

int b; // 局部变量b的声明

b = 20; // 局部变量b的初始化

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

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

// 如果在使用之前不初始化局部变量，编译器会报错

// int c;

// System.out.println("c = " + c);

}

}

以上实例中我们声明并初始化了两个局部变量 a 和 b，然后打印出它们的值。注意，如果在使用局部变量之前不初始化它，编译器会报错。

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

package com.runoob.test;

public class Test{

public void pupAge(){

int age = 0;

age = age + 7;

System.out.println("小狗的年龄是: " + 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:

小狗的年龄是: 7

在下面的例子中 age 变量没有初始化，所以在编译时会出错：

package com.runoob.test;

public class Test{

public void pupAge(){

int age;

age = age + 7;

System.out.println("小狗的年龄是 : " + 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