# java中static关键字的用法

static是静态修饰符，在java中，static可以作用在变量、方法、类上，以及还可以在类中独立成一个静态块。被static修饰的，一般都是属于类本身的，在类加载的过程中，静态的变量就会被初始化，static静态块就会被执行，静态的方法加载到内存，关于静态的东西会单独放到一段内存中，这样对于类的多个实例对象来说，访问静态的东西，其实访问的都是同一块内存空间。

## 静态变量

按照是否静态,对成员变量可以分为两种：一种是被static修饰的变量，叫静态变量或类变量；另一种是没有被static修饰的变量，叫实例变量。两者的区别是：

对于静态变量在内存中只有一个拷贝（节省内存），JVM只为静态分配一次内存，在加载类的过程中完成静态变量的内存分配，可用类名直接访问（方便），当然也可以通过对象来访问（但是这是不推荐的）。

对于实例变量，每创建一个实例，就会为实例变量分配一次内存，实例变量可以在内存中有多个拷贝，互不影响（灵活）。

## 静态方法

静态方法可以直接通过类名调用，任何的实例也都可以调用，因此静态方法中不能用this和super关键字，不能直接访问所属类的实例变量和实例方法(就是不带static的成员变量和成员成员方法)，只能访问所属类的静态成员变量和成员方法。

**静态方法只能访问静态成员变量和静态成员方法，非静态方法可以访问非静态成员方法和非静态成员变量，也可以访问静态成员变量和静态成员方法**。

## 静态类

**一个普通类是不能声明为静态的，只有内部类才可以**。内部类被声明为静态的，就可以直接像普通类一样被访问。

package test;

import test.OuterClass.InnerClass;

public class Test {

public static void main(String[] args) {

// OuterClass.InnerClass innerClass = new OuterClass.InnerClass();

InnerClass innerClass = new InnerClass();

innerClass.a();

}

}

class OuterClass{

static class InnerClass{

public void a(){

System.out.println("OuterClass的内部类InnerClass的方法a");

}

}

}

## 静态块

static代码块也叫静态代码块，是在类中独立于类成员的static语句块，可以有多个，位置可以随便放，它不在任何的方法体内，JVM加载类时会执行这些静态的代码块，如果static代码块有多个，JVM将按照它们在类中出现的先后顺序依次执行它们，每个代码块只会被执行一次。

package test;

public class Test {

private static String str;

static{

Test test = new Test();

test.a();

}

static{

str="1";

System.out.println(str);

}

static{

str="2";

System.out.println(str);

}

public void a(){

System.out.println("Test中的a方法");

}

public static void main(String[] args) {

}

}

执行结果：

Test中的a方法

1

2