

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。构造方法和它在类的**名字相同**，但构造方法**没有返回值**，但不能用void声明构造函数。

不管你与否自定义构造方法，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，它把**所有成员初始化为0**。

可变参数

```
typeName... parameterName
```

在方法声明中，在指定参数类型后加一个省略号(...)。

一个方法中只能指定一个可变参数，它必须是方法的最后一个参数。任何普通的参数必须在它之前声明。

静态方法和实例方法

类方法可以通过类名调用，实例方法不能通过类名调用。

类方法不能被覆盖。不存在多态。

在类方法中可以通过创建实例对象调用类的实例方法。

方法的调用

3. 方法与方法之间的调用

在一个方法内部调用其他方法。

(1) 静态方法内部调用其他方法

如果在本类当中，静态方法可以直接调用静态方法，如果不在一个类当中，静态方法调用其他类中的静态方法，必须通过：类名.静态方法

() ；

```
public class InvokeMethod{
    public static void main (String[] args){
        t2();
    }
    public static void t2(){
        System.out.println("static t2...");
    }
    public static void t1(){
        //静态方法调用非静态方法需必须通过对象来调用
        //InvokeMethod in =new InvokeMethod();
        //in.t2();
        t2();
        System.out.println("static t1");
    }
}
```

如果在不同类当中，静态方法调用其他类的非静态方法，需要导入该类中的包，以及通过创建对象调用。

（2）非静态方法内部调用

在本类中，非静态方法可以直接调用静态方法与非静态方法。

在不同类中，非静态方法调用其他类的静态方法，需要通过导入该类中的包，并且需要通过类名来调用

在不同类中，非静态方法调用其他类的非静态方法时，需要导入该类中的包，还需要通过创建对象来调用。

