

Chapter 6 方法

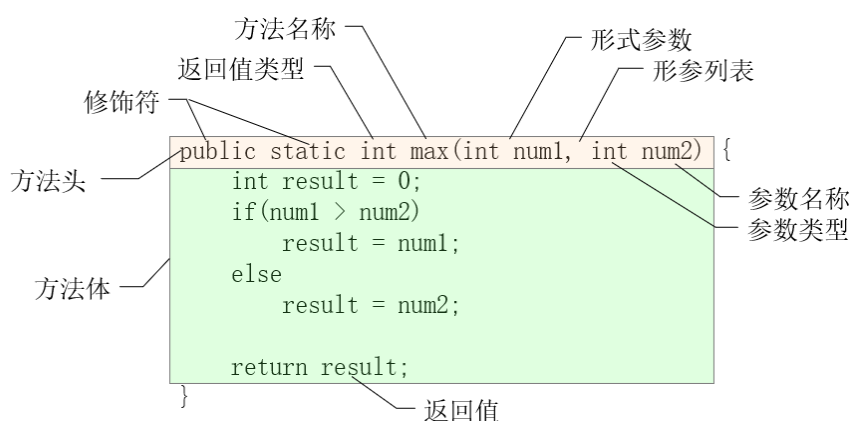
6.1 方法的定义

定义：方法（method）是为执行一个复杂操作组合在一起的语句集合。一个类中可以声明多个方法

语法：采用BNF（巴科斯范式）描述

方法定义示例：

方法定义示例



注意：

- 一个类中不能包含方法签名相同的多个方法，方法签名指方法名称+形参列表（不含返回类型）
- 方法中形参可以用`final`修饰，表示方法内部不允许修改该参数
- 形参不允许有默认值，最后一个形参可以为变长参数
- 方法里不允许定义`static`局部变量
- 方法可以有一个返回值，也可以没有返回值，但构造函数没有返回值，也不能加`void`

6.2 方法调用

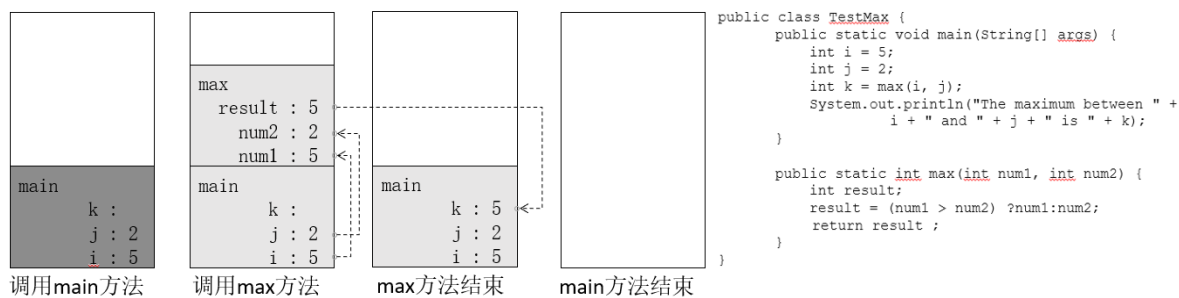
6.2.1 静态方法和实例方法

方法分为静态方法和实例方法。

- 实例方法：依靠实例而存在的方法
 - 调用当前类中的实例方法：通过 方法名 或者 通过`this.方法名`
 - 调用其它类中的实例方法：必须实例化其他类，再通过这个 实例.方法名 调用
- 静态方法：依靠类而存在的方法，也可以通过实例访问
 - 调用当前类中的静态方法：通过 方法名 或者 通过类名.方法名
 - 调用其他类中的静态方法：通过 类名.方法名 或者通过 对象名.方法名

6.2.2 调用堆栈

每当调用一个方法，系统将方法参数、局部变量存储在一个内存区域中，该内存区域称之为调用堆栈，当方法结束返回给调用者时，系统会自动释放想用的调用栈。



6.3 方法参数传递

- 当形参为call by value类型，在方法内修改形参不会影响到实参
- 当形参为call by reference类型，在方法内修改形参会影响到实参
- 实参类型、顺序必须和形参类型、顺序一致
- 类型兼容：子类实参可以传递给父类形参

6.4 方法的重载（Overload）

方法重载指方法名称相同，但形参列表不同的方法，仅返回类型不同的方法不是合法的重载，一个类中可以包含多个重载的方法。

歧义调用：调用时匹配成功的方法可能多于一个，则会产生编译二义性

```
public class AmbiguousInvocation {
    public static void main(String[] args) {
        System.out.println(AmbiguousInvocation.max(1,2));
        // 编译器报错：对max的引用不明确，无法确定用哪个函数，因为参数都能相容
    }

    public static double max(int num1, double num2){
        return (num1 > num2)?num1:num2;
    }

    public static double max(double num1, int num2){
        return (num1 > num2)?num1:num2;
    }
}
```