Atitit.编程语言原理---方法重载的实现与设计 调用方法的原理

[1. 重载包括：普通方法的重载和构造方法的重载 1](#_Toc8885)

[1.1. 横向重载”和“纵向重载” 1](#_Toc12186)

[1.2. 方法签名通过 方法名称，参数数量，参数类型+返回类型 参数顺序（命名参数）来组成 2](#_Toc17118)

[1.3. Atitit 只有返回值不同的方法重载各有利弊(一般的语言设计不支持返回值不同的方法重载） 2](#_Toc7720)

[1.4. 重载（重新载选方法流程，如下图所示 2](#_Toc17469)

[2. 重载的实现方法：：argus参数 vs 默认值。可选参数法 可选参数 4](#_Toc1171)

[2.1. \_\_call()函数是php类的默认魔法函数 4](#_Toc23684)

[2.2. Java通过反射调用方法重载 执行方法 5](#_Toc14280)

[2.3. C++的重载 5](#_Toc13030)

[3. 参考 5](#_Toc14402)

[4. paip.编程语言方法重载实现的原理及python,php,js中实现方法重载 5](#_Toc11195)

# 重载包括：普通方法的重载和构造方法的重载

 所谓重载，就是重复使用。而这个重复使用并不是反复使用同一个一模一样的东西，而是只是借用同一个名字而已。根据这个名字，根据不同的需求，提供不一样的功能。这就是重载。说一个例子，“父亲”这个词，其实是重载了的。如果你叫父亲，那是指你的父亲，而如果别人叫父亲，那是别人父亲。这也是一个重载。根据叫的人不同，也就是调用这个名称的对象不一样，产生的意义就不一样，这就是重载。所以，这里说的就是广义上的重载的概念。

## 横向重载”和“纵向重载”

 清楚了重载的本质分析，再来看看，“横向重载”和“纵向重载”的概念。函数特征标形式是在一个类中实现的多个函数版本，相当于在一个类中平铺多种形式，因此可以理解成横向（平铺）方式。而虚函数形式，是通过继承，构成一个垂直的关系。比如父亲跟儿子，属于两代人，是一个垂直的传递关系，父亲在上一层，儿子在下一层。也就是一个纵向的走向。C++中支持了虚拟函数，提供了重载虚函数的功能，因此就是我们说的“纵向重载”。

## 方法签名通过 方法名称，参数数量，参数类型+返回类型 参数顺序（命名参数）来组成

## Atitit 只有返回值不同的方法重载各有利弊(一般的语言设计不支持返回值不同的方法重载）

不过重载的策略中不包含返回值。

18.1.2 选言推理：为什么重载策略不能包含返回值 - shyleoking的专栏 - 博客频道 - CSDN.NET.html

## 重载（重新载选方法流程，如下图所示

首先得到方法名称，查找符号表，没有跑出ex

通过参数数量判断

通过参数类型判断（强类型语言）

If(nomather()) ex

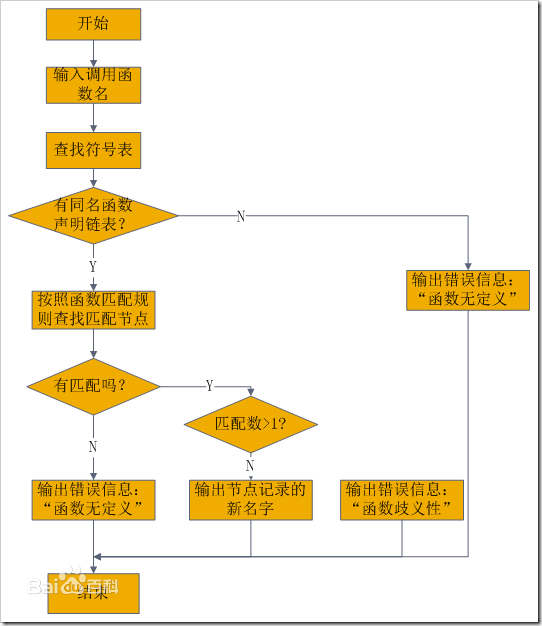
判断参数数量，不同，则ex

If(matchCount>1) ex cant select amigo

方法名和参数数量都对上。匹配参数类型。。

一旦精确匹配就执行精确匹配的方法。如果不能精确匹配。使用简单的knn分类法判别

最接近的方法，比如(int，obj)就比obj,obj 根据精确。。



作者:: 绰号:老哇的爪子 （ 全名：：Attilax Akbar Al Rapanui 阿提拉克斯 阿克巴 阿尔 拉帕努伊 ） 汉字名：艾龙，  EMAIL:1466519819@qq.com

转载请注明来源： http://blog.csdn.net/attilax

# 重载的实现方法：：argus参数 vs 默认值。可选参数法 可选参数

Php通过func\_num\_args();js通过argus参数

Python通过默认值

一个语言选择重载或者可选参数两者中的一种就是了。像C# 4这样同时支持这两种形式的，就比较容易混淆。

著作权归作者所有。

Python的parameter设定也足以应付绝大多数需求了，所以社区**基于效率与语言复杂度的考虑**，干脆就不支持function overloading了

## \_\_call()函数是php类的默认魔法函数

，\_\_call() 在一个对象的上下文中，如果调用的方法不存在的时候，它将被触发:  
<?php  
class MethodTest {  
     public function \_\_call($name, $arguments) {  
        // Note: value of $name is case sensitive.  
        echo "Calling object method '$name' "  
             . implode(', ', $arguments). "\n";  
     }  
}

## Java通过反射调用方法重载 执行方法

paip.编程语言方法重载实现的原理及python,php,js中实现方法重载 - attilax的专栏 - 博客频道 - CSDN.NET.htm

## C++的重载

C语言中函数符号名和对应的函数名是一样的，而C++为了支持函数重载，符号名是在对应的函数名上改编的。如下图所示，函数名为func，而对应的符号名为\_Z4funcv。

实际上函数的编译符号是根据函数名，函数的参数表（包括参数类型和数量）相关的。而且不同的编译器的命名规则不一样。只要能保证相同的函数名和不同的函数参数列表生成的符号名不一样就行

# 参考

PHP-PHP重载和多态的实现？ - 德问 编程社交问答.htm

java怎样实现重载一个方法 - 郑树恒的个人空间 - 开源中国社区.htm

C++重载：横向重载（静态重载）与纵向重载（动态重载）-C++语言-编程语言-C++技术网.htm

为什么 Python 不支持函数重载？ - 编程语言 - 知乎.htm

Atitit “默认参数”和“方法重载”同时支持的话有二义性的问题

Atitit 只有返回值不同的方法重载各有利弊

# [paip.编程语言方法重载实现的原理及python,php,js中实现方法重载](http://blog.csdn.net/attilax/article/details/22698945)

atiend