Attit 现代编程语言重要特性

目录

[第一章 可读性与开发效率 简单性 1](#_Toc11575)

[第一节 垃圾回收 1](#_Toc4888)

[第二节 自动垃圾回收 1](#_Toc20717)

[第三节 命名参数 函数多返回值 1](#_Toc17847)

[第四节 多语言 语言交互性 2](#_Toc9835)

[第五节 匿名函数和闭包 2](#_Toc10566)

[第六节 流式样接口 vs wrap 2](#_Toc19263)

[第七节 文本块字符串模式 2](#_Toc22889)

[第八节 Record轻量级类 2](#_Toc14236)

[第二章 稳定性方面 2](#_Toc5148)

[第一节 整除函数 — 安全的除法 (即使是被 0 整除) 2](#_Toc9667)

[第三章 性能方面 2](#_Toc5641)

[第一节 jit 2](#_Toc17782)

[第二节 并发编程TPL 2](#_Toc9890)

[第三节 直接执行源码 2](#_Toc9263)

[第四节 强类型机制 可选 2](#_Toc14854)

[第四章 Ref 3](#_Toc14329)

[第一节 Php新特性 java新特性 3](#_Toc32494)

[第二节 C# 新特性 js 新特性 3](#_Toc10772)

[第三节 Go 语言最主要的特性： 3](#_Toc8512)

# 可读性与开发效率 简单性

## 垃圾回收

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 垃圾回收 | √ | × | × | √ | √ | √ | × | √ |
| 多重继承 |  |  |  |  |  |  |  |  |

## 自动垃圾回收

## 命名参数 函数多返回值

## 多语言 语言交互性

## 匿名函数和闭包

## 流式样接口 vs wrap

i=Convert.ToInt(Convert.ToFloat(Convert.ToDouble(Convert.ToInt(i))));  
这样很不直观，如果有扩展方法的话就为  
i=i.ToInt().ToDouble().ToFloat().ToInt();

## 文本块字符串模式

## Record轻量级类

# 稳定性方面

## 整除函数 — 安全的除法 (即使是被 0 整除)

该函数会返回第一个参数被第二个参数整除后结果的整数部分。当除数（也就是第二个参数）值为 0 时，该函数会抛出一个 E\_WARNING 的错误并且返回 FALSE。

# 性能方面

## jit

## 并发编程TPL

## 直接执行源码

## 强类型机制 可选

，但是只有系统类型，没有用户类型。。比如user等。。

"强类型机制"是指在声明变量的时候，语言的规范强制程序员编程的程度。在"强类型机制"类型的语言中(大部分现代语言都支持强类型机制)，每一个数据的取值和使用的对象都必须从属于对上下文来说合适的类型。静态类型语言在编译时使用类型检查强制类型转换。动态类型语言在运行时检查类型，

多重继承最小化代码复制的数量，进而减少可能引起维护问题的不一致性，所以很重要。在静态类型语言中，多重继承可能允许新的类主动位于所有超类之下，这减少了系统其他部分需要的特例编程的数量。

# Ref

## Php新特性 java新特性

Atitit php php7.0 5.0 5.3 5.4 5.5 -6.0的新特性总结与比较

## C# 新特性 js 新特性

## Go 语言最主要的特性：

* 自动垃圾回收
* 更丰富的内置类型
* 函数多返回值
* 错误处理
* 匿名函数和闭包
* 类型和接口
* 并发编程
* 反射
* 语言交互性

Atitit java 新特性10 11 12 13 14 15