Atitit 提升开发效率总结

目录

[1. declara dynamic Dsl化 fp script 1](#_Toc775)

[1.1. 各种语法新特性 linq等 2](#_Toc24465)

[1.2. duck typing。 2](#_Toc2687)

[1.3. 静态语言需要自己实现一个vm实现dynamic概念。。 2](#_Toc8644)

[1.4. Java+js 解析器内部 2](#_Toc18102)

[2. Tool 集成一体化ide化 2](#_Toc16080)

[2.1. Ember的web server 2](#_Toc27682)

[2.2. Ide embed db manager tool 2](#_Toc1019)

[2.3. Snippet 2](#_Toc31475)

[3. Lib framework类库化 2](#_Toc16560)

[3.1. Attilax sdk 2](#_Toc31995)

[3.2. Dev plat 3](#_Toc14149)

[3.3. cli集成 3](#_Toc13274)

[3.4. Hybreid 3](#_Toc4337)

[3.5. 集成gui比如flashfxp 3](#_Toc16794)

[4. Unit 3](#_Toc22143)

[4.1. Mini envi spring mybatis dubbo 3](#_Toc26402)

[5. 开发方法 oo ori lib 3](#_Toc15227)

[5.1. 提升复用级别 3](#_Toc3865)

[5.2. 培训 3](#_Toc14320)

[5.3. 多屏幕，，打印chu来文档 use cp as another screen 3](#_Toc20804)

[6. 元编程 3](#_Toc3313)

[7. Ref 4](#_Toc14683)

# declara dynamic Dsl化 fp script

动态语言和静态语言各有一些优势，这也是两个阵营争论多年的内容。不过Anders认为它们各自都有十分重要的优点，而未来不属于其中任何一方。他表示，从编程语言发展过程中可以观察到两种特点正在合并的趋势，未来应该属于两者的杂交产物

## 各种语法新特性 linq等

## duck typing。

## 静态语言需要自己实现一个vm实现dynamic概念。。

## Java+js 解析器内部

# Tool 集成一体化ide化

同样类似，现在的IDE包含了无数强大的功能，例如语法提示，重构，调试器等等。与

## Ember的web server

## Ide embed db manager tool

## Snippet

现在有很多编辑器都有snippet。即将你常用的代码写入相应语言的snippet文件中，在使用时只需敲几个字母就能生成一大段代码。这种方法可以说是以简驭繁。但是这些代码段应该经常整理，避免最后连自己都忘记了。

# Lib framework类库化

## Attilax sdk

。Anders认为编程语言的发展非常缓慢，期间当然出现了一些东西，例如面向对象等等，但是远没有好上1000倍。事实上，近几十年来的努力主要体现在框架及工具等方面（如下图）

## Dev plat

## cli集成

## Hybreid

## 集成gui比如flashfxp

# Unit

## Mini envi spring mybatis dubbo

# 开发方法 oo ori lib

## 提升复用级别

## 培训

## 多屏幕，，打印chu来文档 use cp as another screen

# 元编程

“元编程”实际上是“代码生成”的一种别称，其实在日常应用中我们也经常依赖这种做法。观察动态语言适合元编程的原因也是件十分有趣的事情。

动态语言的关键之一便是“元编程”，“元编程”实际上是“代码生成”的一种别称，在日常应用中开发人员其实经常依赖这种做法了。在某些场景下使用动态语言会比静态语言更加自然一些。例如在C#或Java里使用ORM时，一种传统做法是让代码生成器去观察数据库，并生成一大堆代码，然后再编译。而动态语言并没有编译期和执行期的区别，例如在Ruby on Rails中使用ActiveRecord便无须定义各式字段。

Anders谈到，他和他的团队也在努力改进静态语言的元编程能力，如他们正在实现的“编译器即服务（Compiler as a Service）”。传统的编译器是一个黑盒，一端输入代码，而另一端便会生成.NET程序集等数据，开发人员很难参与或理解它的工作。但是在很多时候，开发人员并不一定需要编译器来生成程序集，他们需要的是一些树状的表现形式，然后对它进行识别和重写。因此，开发人员可能会越来越需要一些开放编译器功能的API。这么做可以让静态类型语言获得许多有用的功能，包括元编程以及可操作的完整对象模型等等。

# Ref

编程语言的发展趋势及未来方向 - CSDN博客.html

为什么net比java效率高开发