**Scala**

|  |  |
| --- | --- |
| **审核人** |  |
| **重要性** | 中 |
| **紧迫性** | 中 |
| **拟制人** | 张包峰 |
| **提交日期** |  |

**作者：张包峰**

(版权所有,翻版必究)

**修改记录**

|  |  |  |
| --- | --- | --- |
| **更新时间** | **变更内容** | **变更理由** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

[**作者：张包峰** 1](#_Toc394233764)

[1.1 Scala School 4](#_Toc394233765)

[1.2 Effective Scala 4](#_Toc394233766)

[1.3 Scala In Action 4](#_Toc394233767)

[1.4 Scala In Depth 4](#_Toc394233768)

## Scala School

## Effective Scala

## Scala In Action

## Scala In Depth

### 基本属性

Scala的特点：

函数式编程和面向对象编程；

富有表达力的语法和静态类型；

高级的语言特性和丰富的java结合

函数式编程重在结合和操作verbs。视为bottom-up风格。

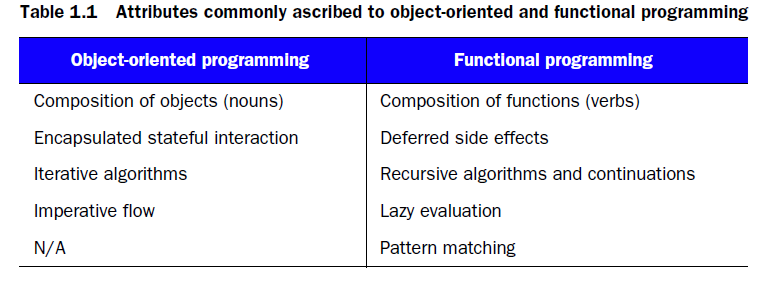
把软件解耦为需要执行的行为、动作。Functions带有数学概念，纯操作input输入，所有的变量是不可变的(对并发有帮助)。

函数式编程更多地利用类型系统(type system)来描述function做的事情。

面向对象编程更强调nouns，并附加上动词。视为top-down风格。

对象拥有属性，行为和状态。对象之间的交互，service class。

两者看待“软件”的视角不同。



静态类型和表达式的四个特点：

改变了类型声明的位置；

类型推断；

可扩展语法；

Implicits

关于Implicits：

Scala.Predef对象

Implicits的作用介绍

HotSpot的几个特性：

Method inlining;

On Stack Replacement(OSR);

Escape Analysis;

Dynamic deoptimization

### Core Rules

REPL带来的Experiment-driven development

### 使用implicit

### type system

### 使用implicit和type system

### Collections

三方面看：

不可变和可变集合；

eager和delayed evaluation；

sequential和parallel evaluation

### Actors

## Scala 类和语法

AnyRef相当于java里的Object

Scala.Predef对象

带来Commonly Used Types，Console I/O，Assertions，Implicit Conversions

classOf[T]相当于java的T.class