

编程语言

面向对象编程

函数式编程

Haskell

✎ 修改

关注者


203

被浏览

10,448

# 函数式编程如何优雅的处理很多 多个函数都要用到的 参数? ✎ 修改

他们也关注了该问题



面向对象解决的一个比较重要的问题就是很好的封装了 global constant (variable) 这样子保证了每一个函数中的参数不会过长。 比如我现...显示全部

关注问题

✎ 写回答

+ 邀请回答

💬 添加评论

🚩 分享

★ 邀请回答

🚩 举报

...

查看全部 9 个回答



**Felis sapiens**   
函数式编程、编程语言、编程 话题的优秀回答者

草莓大福、hhhhhhhhh、祖与占等 33 人赞同了该回答

1. 设计一个 Config 数据类型，包含需要用到的配置。然后所有需要用到该配置的函数，显式传 Config 作为函数参数，为方便起见，通常设为第一个参数。函数大概长这样：

```
work :: Config -> IO ()
```

2. 一个函数实际上就是一个 reader monad。如果函数带副作用，且使用 mtl 管理副作用，那么“读取配置”显然也算是一项副作用，可以用 MonadReader 来表示，在函数体中可以使用 ask 获取当前配置，使用 local 修改配置并重启计算。函数大概长这样：

```
work :: (MonadReader Config m, MonadBase IO m) => m ()
```

3. 如果项目复杂，一个 Config 类型不足以用在所有场合，需要多个 Config，可以在 2. 的基础上使用 classy lens 对 config 类型进行抽象。比如某个函数需要从配置中读取 SessionId，那么先实现一个 HasSessionId class，提供从某个类型聚焦到 SessionId 的 lens（也就是 getter/setter 函数），然后函数这么写：

```
work :: (MonadReader env m, HasSessionId env, MonadBase IO m) => m ()
```

现在 work 函数可以传入不同种类的 Config 类型调用，只要这些 Config 类型包含 SessionId。

4. 在函数式编程中，题主提出的这个问题叫“configuration problem”。Oleg Kiselyov 提出过一个很有意思的解决方案：Functional pearl: implicit configurations，实现参考 reflection: Reifies arbitrary terms into types that can be reflected back into terms

5. 当然，最方便的还是用全局变量传配置了，没有全局变量的编程语言还好意思出来混？  
cambridge.org/core/jour... 我觉得这个也很优雅。

编辑于 2017-04-16

▲ 赞同 33 ▼

💬 3 条评论

🚩 分享

★ 收藏

❤ 感谢

...

广告 X

腾讯云新注册用户域名抢购1元起

.com

.cn

.xyz

腾讯云新用户域名抢购1元起

.xyz 更有多种爆款组合选购

➔

关于作者



**Felis sapiens**

✎ 函数式编程、编程语言、编程 话题的优秀回答者

👤 电影旅行敲代码、Antokha Yuuki、暮无井见铃也关注了她

回答

624

文章

40

关注者

14,871

已关注

💬 发私信

被收藏 14 次

编程语言与编译原理 5,338 人关注  
酿酿酿酿酿泉 创建

计算机 3 人关注  
魏王雅望非常 创建

PL 2 人关注  
lsdsjy 创建

IT 1 人关注  
那罗延 创建



更多回答



祖与占

函数式编程 话题的优秀回答者

14 人赞同了该回答

好问题, 这个问题一直放在我的 problem list 里面, 虽然有 Oleg Kiselyov 这种问题终结者在这上面发了 paper, 还有 Edward Kmett 根据 paper 写了个库, 但是这 paper 一直没认真看, 库也没用过, @Canto Ostinato 也已经介绍得差不多了, 所以没我什么事了 (逃

其实写 Java 的人对这个问题最熟悉不过了, 这就是依赖注入最常见的一个应用场景, 例如写 Web 服务需要连数据库打 log 之类的, 拼命把数据库连接跟 logger 引用传来传去肯定蛋疼, 所以把数据库连接/ logger 看做是种依赖注入到代码用起来就比较方便.

Haskell 社区在这方面也不少探索:

例如这篇 [The Service Pattern](#), 搞得写 Haskell 好像写 ML 一样 (

[A Solution to the Configuration Problem in Haskell](#) 里提到的一些方案

Using mutable references and some hacking with [unsafePerformIO](#), which always gives the programmer a bad conscience.

[展开阅读全文](#)

赞同 14

3 条评论

分享

收藏

感谢

...



dboy

itlr.cc

3 人赞同了该回答

首先, 你需要的是一个“优化”, 而不是“设计”, 设计应该是直观明了的, 函数形参该怎么设计怎么设计, 不应该为了“优化”而影响“设计”。这是大的原则。

如果你用 Javascript, 通常会用到 bind 来做优化 (这个过程也被叫做 inversion of control)。

比如有一个比较一般的 process 函数

```
function process(config) { ... }
```

假设另有两个比较具体的 foo 和 bar 函数, 也要用到 config, 但同时也用到别的参数

```
function foo (config, x) { ... }
```

```
function bar (config, y, z) { ... }
```

你可以

```
var process foo = foo.bind(this, config)
```

[展开阅读全文](#)

赞同 3

1 条评论

分享

收藏

感谢

...

[查看全部 9 个回答](#)

llvm

bob 创建

0 人关



## 相关问题

[如何向一个只有命令式编程背景的人解释函数式编程中的函数?](#) 14 个回答

[如何使用 C++ 实现一个 and 函数?](#) 3 个回答

[函数式编程的函数是如何实现的?](#) 11 个回答

[函数内部的变量在函数执行完后就销毁, 为什么可变对象却能保存上次调用时的结果呢?](#) 8 个回答

[C++ 为什么要让 struct 可以定义成员函数?](#) 8 个回答

## 相关推荐



森懂物理学：理解世界的极简指南

共 31 节课

[试听](#)



法式优雅购物指南

王岚

★★★★★ 752 人参与



函数式编程思维

Neal Ford

143 人读过

[阅读](#)

广告 X

腾讯云新注册用户域名抢购1元起

.com

.cn

.xyz

腾讯云新用户域名抢购1元起



更有多种爆款组合选购



刘看山 · 知乎指南 · 知乎协议 · 隐私政策

应用 · 工作 · 申请开通知乎机构号

侵权举报 · 网上有害信息举报专区

违法和不良信息举报: 010-82716601

儿童色情信息举报专区

电信与服务业务经营许可证

网络文化经营许可证

联系我们 © 2018 知乎

