链滴

函数式编程

作者：MessiahJK
原文链接：https://ld246.com/article/1538071197739
来源网站：链滴

一.含义

使用纯函数编写程序

函数式编程是一种编程范式，构建计算机程序的方法，把计算当作数学函数求值的过程，避免了改变态和可变的数据

二.重要概念

纯函数（Pure Function），或函数的纯粹性（Purity），没有副作用（Side Effect）

纯粹性（即无副作用）；引用透明

副作用：状态的变化（mutation）（修改全局变量，抛出异常，IO读写，调用有副作用的函数）

例子：

```
var x=1
def XplusY_V1(y:Ing) =x+y
def XplusY_V2(y:Ing) ={x+=y;x}
x
>res0:Int=1
XplusY_V1(2)
>res1:Int=3
x
>res2:Int=1

XplusY_V2(2)
>res3:Int=3
x
>res4:Int=3
```

XplusY_V1不具备副作用，而XplusY_V2具备副作用

XplusY_V1是个纯函数，XplusY_V2不是

引用透明（Referential Transparency）

对于相同的输入，总能得到相同的输出

如果f(x)的参数x和函数体都是引用透明的，那么函数f是纯函数

违反引用透明的案例

```
var x=new StringBuilder("Hello")
>x:StringBuider=Hello
var y=x.append(" World!")
>y:StringBuider=Hello World!
var z=x.append(" World!")
>x:StringBuider=Hello World! World!
```

对于相同的输入，append的返回是不一样的，append方法违反了引用透明性

不变性（Immutability）

为了获得引用透明性，任何值都不能变化

函数第一（First-class Function）
一切都是计算，函数式编程中只有表达式，变量、函数都是表达式

高阶函数（Higher order Function）

闭包（Closure）

表达式求值策略：严格求值和非严格求值
Call By Value & Call By Name

惰性求值（Lazy Evaluation）
当定义表达式的时候不会立即求值，只有第一次用到表达式才会去求值

递归函数（Recursive Function）
函数式编程没有循环语句，所有的循环以递归来实现

尾递归（Tail Recursion）
用来解决递归函数堆栈溢出的问题

三.优点

1.生产效率高
同样功能的程序，lisp代码长度可能只是C代码的1/7~1/10
——硅谷创业之父Paul Graham 《黑客与画家》
(注：lisp是历史上上第一门函数式编程语言)

2.易于推理（Reasoning）
构造完程序之后对于给定的输入总是能得到确定的输出，给程序的调试带来了方便性

3.并行编程
多核计算/云计算

4.便于管理

5.代码的热升级

函数式编程没有副作用，只要保证接口不变，内部实现是外部无关的。所以，可以在运行状态下直接
级代码，不需要重启，也不需要停机。