

函数式编程与面向对象编程的比较

面向对象的优点：

面向对象程序设计可以看作一种在程序中包含各种独立而又互相调用的对象的思想，这与传统的思想刚好相反。传统的程序设计主张将程序看作一系列函数的集合，或者直接就是一系列对电脑下达的指令。面向对象程序设计中的每一个对象都应该能够接受数据、处理数据并将数据传达给其它对象，因此它们都可以被看作一个小型的“机器”，即对象。目前已经被证实的是，面向对象程序设计推广了程序的灵活性和可维护性，并且在大型项目设计中广为应用。此外，支持者声称面向对象程序设计要比以往的做法更加便于学习，因为它能够让人们更简单地设计并维护程序，使得程序更加便于分析、设计、理解。同时它也是易拓展的，由于继承、封装、多态的特性，自然设计出高内聚、低耦合的系统结构，使得系统更灵活、更容易扩展，而且成本较低。

在面向对象编程的基础上发展出来的23种设计模式广泛应用于现今的软件工程中，极大方便了代码的书写与维护。创建型模式，共五种：工厂方法模式、抽象工厂模式、单例模式、建造者模式、原型模式。结构型模式，共七种：适配器模式、装饰器模式、代理模式、外观模式、桥接模式、组合模式、享元模式。行为型模式，共十一种：策略模式、模板方法模式、观察者模式、迭代子模式、责任链模式、命令模式、备忘录模式、状态模式、访问者模式、中介者模式、解释器模式。

面向对象编程的缺点

面向对象编程以数据为核心，所以在多线程并发编程中，多个线程同时操作数据的时候可能会导致数据修改的不确定性。在现在的软件工程中，由于面向对象编程的滥用，导致了很多问题。首先就是为了写可重用的代码而产生了很多无用的代码，导致代码膨胀，同时很多人并没有完全理解面向对象思想，为了面向对象而面向对象，使得最终的代码晦涩难懂，给后期的维护带来了很大的问题。所以对于大项目的开发，使用面向对象会出现一些不适应的情况。面向对象虽然开发效率高但是代码的运行效率比起面向过程要低很多，这也限制了面向对象的使用场景不能包括那些对性能要求很苛刻的地方。

函数式编程的优点

在函数式编程中，由于数据全部都是不可变的，所以没有并发编程的问题，是多线程安全的。可以有效降低程序运行中所产生的副作用，对于快速迭代的项目来说，函数式编程可以实现函数与函数之间的热切换而不用担心数据的问题，因为它是以函数作为最小单位的，只要函数与函数之间的关系正确即可保证结果的正确性。函数式编程的表达方式更加符合人类日常生活中的语法，代码可读性更强。实现同样的功能函数式编程所需要的代码比面向对象编程要少很多，代码更加简洁明晰。函数式编程广泛运用于科学研究中，因为在科研中对于代码的工程化要求比较低，写起来更加简单，所以使用函数式编程开发的速度比用面向对象要高很多，如果是对开发速度要求较高但是对运行资源要求较低同时对速度要求较低的场景下使用函数式会更加高效。

函数式编程的缺点

由于所有的数据都是不可变的，所以所有的变量在程序运行期间都是一直存在的，非常占用运行资源。同时由于函数式的先天性设计导致性能一直不够。虽然现代的函数式编程语言使用了很多技巧比如惰性计算等来优化运行速度，但是始终无法与面向对象的程序相比，当然面向对象程序的速度也不够快。

函数式编程虽然已经诞生了很多年，但是至今为止在工程上想要大规模使用函数式编程仍然有很多待解决的问题，尤其是对于规模比较大的工程而言。如果对函数式编程的理解不够深刻就会导致跟面相对象一样晦涩难懂的局面。

总结

函数式编程和面向对象编程各有利弊，一个语法更加自由，一个健壮性更好。作为程序员应该对两种编程方式都有所了解，不管是哪种方式，只要能够很好的解决当前的问题就是正确的方式，毕竟对于软件工程来说解决问题是最主要的，用的工具反而没有那么重要，就像对程序员来说语言不重要，重要的是解决问题的思想。现在这两者的发展趋势是相互借鉴的，许多以面向对象作为基础的语言例如Java等都在新的版本中添加了对函数式编程的支持，而函数式编程则借鉴了一些在面向对象语言里用的一些编译技巧使得程序运行更快。