

论C#变得越来越臃肿是不可避免的 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/245/2021_2022__E8_AE_BA C_23_E5_8F_98_E5_BE_c97_245978.htm C# 2.0发布的时候，我们回头看Java，总认为这个语言怎么发展得这么慢?但当C#发展到3.0的时候，它也开始显示出臃肿之态了，这是否会也会带来什么连锁效应呢?6年前，我是个Java的拥护者，当时C#还是1.0版，我经常和师傅争论Java如何比C#好，于是他给我一个回答：“我们的COM比Java早了近5年，所以我们更成熟.我们的.NET比Java晚了5年，所以更先进”。虽然这么比较有“偷换概念”的感觉，但现在想想其实有另一层意思“成熟与先进”的矛盾。Lisp、Haskell、Scheme这些语言也都可以被称之为“伟大”，但为什么很少有人去学呢?因为需要用太多的东西“充斥”我们的大脑后才可以使使用。Java和C#之所以可以快速地被普遍接受，一个很重要的原因就是因为他们们的简单与清爽。但当明年春天C# 3.0发布的时候会怎么样呢?虽然你可以将WCF、WF、WCS和WPF视为.NET的外挂，不予理会，但LINQ是个不好回避的内容，因为它在处理数据访问(关系型的、非关系型的)方面有比较明显的优势，所以即便你个人排斥它，其他还是会有很多人用。最后很可能成为这样一种局面：参与到一个项目组，自己只能从事一些表层业务开发，因为下层的公共封装机制都是用LINQ编写的，况且还有Enterprise Library这个“样板工程”在后面催着。可以这么说，C#越来越臃肿是个必然的趋势，作为.NET语言的“主力”，随着新的开发架构的出现，C#的复杂性还会增加，同时很可能导致革新特性越出越慢，毕竟牵扯的内容多了

，作为“主力”除了要考虑语言特性间的协作外，还要充分考虑处理效率。不过比起“一条道跑到黑”的Java而言，.NET平台有个优势CLS(Common Language Specification，公共语言规范)。相信Java的设计者不太愿意，也不敢随便为了一个“快速走红”但还没有2年时间市场考验的技术趋势就去修改Java编译器..NET不同，“C#红旗不倒的同时，.NET平台可以彩旗飘飘”，比如Spec#就是个例子，为了避免null对于软件的影响，.NET家族诞生了Spec#，目的就是通过非null这个前提，提高数据验证、异常处理、堆栈管理的能力，利于开发者提供更高质量的软件.F#也是，虽然C#是强类型的，但动态语言式的开发一样可以基于这个“小兄弟”开发，加上它和其他.NET语言前辈基于同一个CLR环境，所以功能毫不逊色。综上所述，C#臃肿是不可避免的，而且很可能会像Visual C 一样，因为语言的复杂性，导致C#开发人员技术能力的两极分化。但同时，借助试验性.NET语言的支持，即便需要集成新的特性，也不会像某些语言一样从头开始。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com