

程序员的痛苦:先学VB.NET还是C#? PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/139/2021_2022__E7_A8_8B_E5_BA_8F_E5_91_98_E7_c100_139936.htm 在过去的一年中，互联网上的各大讨论区或者电子邮件的讨论列表都对微软的VB.NET以及C#的各种优越性做了探讨。这些讨论围绕的主要问题就是，我应该先学哪一个，VB.NET还是C#？我写这篇文章的目的就是想帮您解决这个问题。我并不是想动摇你倾向哪一种语言而是想解决一些大家在基本问题上的疑惑，以便大家能够作出自己的决定，选择一种自己觉得用起来最舒适的语言。我将尽量避免讨论一些语法上的模棱两可的话，就像“C#的括弧太多了，”“VB.NET句子太冗长，”或者“我讨厌C#（或者VB.NET）因为它能（或者不能）区分大小写。”之类的话。评论语法的好坏是你个人品味的问题。相反，我将着重讨论一些我见到的关于这两种语言的技术方面的讨论。在C#方面作为微软公司最新的一种语言，并且由于它又是Java语言的小翻版，C#引起了广大的关注。人们看上去喜欢一种语言仅仅取决于它是最新的，程序开发者们总是喜欢用最新的工具工作。其它的一些选择使用C#的理由更为具体一些。领导潮流的东西总是无懈可击的“如果我正准备学一门新的语言，我还是应该学C#。”这也许也是你经常听到的言论。那些推理总是这样进行的：“VB6转变到VB.NET变化已经非常大了，以至于它基本上就是一门新的语言。如果我无论如何打算学习新语言，我想还是学C#吧，因为它是特别为.NET类的库设计的。”这也是我听到过的关于这两方面的最苍白的争论。你也可以同样理直气壮的说，如果我

无论如何打算学习新语言,我想还是学VB.NET吧,毕竟它也是一门新的语言。另外,让我们想想为什么VB.NET从其先驱者那里如此激烈地演变到现在的样子:它为了适应.NET类的库而被重新设计了。对比管理过的和没有管理过的代码“C#允许我写那些运行在CLS存储器控制之外的非管理代码,我可以直接访问存储器,并且使用指针。让代码自由地运行,包括使用存储器的管理,可以得到更高的效益。”这个观点有3个问题需要考虑:首先,我们不应该在Beta版本的开发环境下讨论性能问题。举个例子:在.NET的Beta1和Beta2版本之间有显著的管理代码运行速度的改善。第二,我们还不能把非管理代码比管理代码能获取多少利益量化,并且是否值得为了这些好处冒险。可以去看看Eric Gunnerson在MSDN上的这篇文章。第三,尽管VB.NET不能建立非管理代码,它可以通过System.Runtime.InteropServices名字空间的使用,来访问并工作于非管理存储器。C#有内置的XML文件编制器“C#编译器包括直接被嵌入成为源代码的XML文件编制器在内。如果我使用C#,我同时编写了代码并编制了文件。”使用过JavaDoc的人都知道,把你的文件编制加到你的源代码中是多么的有用。源代码和文件编制可以同时更新,因此至少在理论上讲,你的文档永远都不会过时。不过,以我的经验来看,相对少数的Java开发者还是在使用JavaDoc。这样,问题就变成“你将使用它吗?”如果你的对这问题的解答是“是”,你有足够的理由试试C#。关于VB.NET又怎么样呢?在很多真正的开发者看来,VB像玩具语言似的,从某种角度看,也确实是这样的。迄今为止,VB远比我们所知道的那两三个弱点更多。不过VB.NET确实是和C#同样强大的.NET开发

语言。有些人说它更强大。VB.NET有内置的（插入特点）支持；而C#没有“VB.NET内置了很多东西像字符串操作(Mid, InStr, 等等)和类型转换（例如CInt）。C#缺乏这些内置的支持，所以，我所需要的东西，在C#中很难找到。如果你抓住这些你应该Mid或者CInt功能不放，而最终认为这就是VB.NET强于C#的证据，你最好去看看Microsoft.VisualBasic namespace。你将在那里发现大部分VB.NET内部命令和应用功能。这些功能在namespace中被保存之后，任何CLS兼容的语言都能使用他们，就像列表A中所显示的那样。这些例子削弱了我们的争论，不是吗？更好捆绑的支持就是不支持“VB.NET与COM实体的捆绑支持更好一些。”我也只是看到了一点点而已，并且我决定再也不在支持方面作任何推理。从我迄今为止所观察到的，这不是真的。C#和VB.NET必须采用runtime callable的包装以及等量的源代码来执行一个早期的实体。同样地，执行一个晚期的实体也需要相同数量的代码。VB.NET使用IDE中的后台编译如果你不能找到其他的认为VB的开发环境好的例子，你至少不得不承认它的源代码编辑是很有特点的。你能一边打字一边字面上排除你的代码的错误。麻烦就是那些很弱智的编译错误信息框总是弹出来，并且如果你把你的喇叭声音开得过大的话，报错的嘀嘀声也许会吓到你。Visual Studio.NET避免了这种惊吓，直到你修改完成，并且处理了一些消极的错误，提示系统经过了微软的改进：他会在那些错误语句的下面打上弯弯曲曲的下划线。VB.NET背景编译程序/句法检验器非常复杂，而且很客气地指出你的错误。从某些方面看，它能更准确地告诉你如何修改你源代码中的错误。当C#有它自己的语法检查器，并且可

以查出括弧的匹配，计算圆括弧的多少，显示丢失的分号，但是它还是不能像VB.NET那样使用简单。再继续讨论这两种语言的优越性确实会让我心烦的，不过微软的话确实是一个真理，那就是所有的.NET语言都是平等建立的。那些主张C#优于VB.NET的人（反之亦然）和那些攀比工资的开发者们一样错了。我要强调的是，那些有远见的技术公司不再会去寻找具有某种开发语言经验的程序员，而是去寻找那些有.NET类库开发经验的程序员。因此我劝你不要过分的担心自己的选择到底是什么：随便找一个你觉得有兴趣学的语言，认真地学好他的框架结构就行了。如果你最终认为我是错的，并且市场也不要求你一定要选择一种语言，那你就尽管嘲笑我吧。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