计算机二级VB辅导:脆弱的基类计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文
https://www.100test.com/kao_ti2020/645/2021_2022__E8_AE_A1_E7_AE_97_E6_9C_BA_E4_c97_645168.htm“ 既然说是脆弱，当然是指它象蛋壳一样不堪一击喽。这个问题其实很好理解。程序总是由人来设计与编写的，所以工作开始时考虑不到某些问题当然也是很正常的事。所以可能在工作进行了一段时间后发现基类需要变更。你想，如果我在基类中更改了成员的数据类型，以及那些允许重写的那些方法和属性，那派生类及其子类还能正常工作吗？尤其是当一个团队中的多个开发人员一起来创作基类和派生类时，就更是要命了。很多情况下，大家可能已经把基类和一些派生类编译成二进制形式进行提交了。更改基类，重新编译再分布，会牵一发而动全身，导致项目的崩溃。所以我们把这叫做‘ 脆弱的基类 ’。也就是说，它是整个工程中最薄弱最致命的环节。 ” 大李眉头一直紧锁着，想必是回想起了自己受打击的经历。“ 这么严重呀，现在的软件工程设计方法会不会对这个有很好的解决方案？ ” 我努力想缓解一下大李的严肃神情。“ 如果对项目的前期设计考虑尽可能周详，在工程实施中对项目的代码控制与相关性分析做得踏实，会起到很好的效果。但是不管一个人如何努力，有时还是无法避免对基类进行不可预见的更改。我们摸索过很久，有了一些处理的手段。 ”“ 真是成事在人呀，我们现在有什么解决之道？ ” 我也一下子振奋起来了。“ 呵呵，并不是什么完美解决方案。只能在某种程度上减轻危害。我们常用的一个方法，最直接的思想就是，把有可能发生的更改全都放在派生类中进行，不在基类中做

。”“这具体是什么意思呀，我还是不太明白。”我不好意思地挠挠头。大李微笑着点点头，看来是知道我不会明白的了。“我们在基类中使用的是抽象类，它内含的方法与属性只有定义，没有进行实现，而把实现部分都放在派生类中做。这样一来，抽象类自身是无法被实例化的。但是它的好处不言而喻，就是有可能发生的实现上的更改都会只涉及到它的派生类了。VB.NET中就提供了这样的手段。”Public MustInherit Class CBaseHenry Public MustOverride Sub subX(ByVal x As Integer) Public MustOverride Function fcnY(ByVal y As Integer) As Long End Class Public Class CDerivedHenry Inherits CBaseHenry Public Overrides Sub subX(ByVal x As Integer)’写入实现的代码 End Sub Public Overrides Function fcnY(ByVal y As Integer) As Long’写入实现的代码 End Function End Class“这里要注意两个问题，一个是关键字，我们用MustInherit来修饰类名，使类成为抽象类，在它的成员中，把方法和属性前加入MustOverride修饰符表示它们必须在派生类中加以实现。第二个要注意的是，派生类必须对所有用MustOverride标识的基类方法和属性都进行实现，只重写了subX，不写fcnY编译器会报错的。”“这的确可以解决一部分问题，但好象只能解决在基类中进行实现的代码有更改的问题，对于数据类型的更改好象没有什么效果。”我看了好一会，发出了这样的疑问。“所以我刚才说，是在某种程度上进行解决嘛。”大李也不由笑了起来，“不过你提的这个问题，倒不是太麻烦，我们可以在派生类中用Shadows来解决呀！（详见本报上一期《重载与隐藏》）”这倒是个不错的主意，我心中暗暗评价了一番。突然我又想

到一个问题：“ 如果基类要做功能扩展，怎么办呀？ ”“ 如果是要做扩展，最安全的方法是添加新成员，而不是对基类的大肆修改。一般是往派生类添加设计时缺失的新成员。最好不要使用Overloads关键字来命名与基类相同的成员，那样往往会带给你意想不到的问题。最好重新定义新成员，命名上也要尽量与基类已有的成员名区分开来。其实，也可以往抽象类基类中添加新成员的定义，但这样一来，需要为基类制定版本，虽然不会对应用程序造成毁灭性的危害，但是应该要能够完全地控制与管理自己的代码。我们一般是不希望扩展基类的。 ”我已经大意上领会了大李的一片苦心：“ 您的意思，是不是指基类的脆弱问题实际上是客观存在的，我们所做的就是要最大程度的减小这个问题带来的危害？ ”大李眼中闪过一丝赞许的笑意，颌首道：“ 没错，对于一个应用程序的设计者来讲，想使用面向对象方法来开发，必须要在设计的时候精心策划类的层次结构。一般来说，是有这样几个准则需要把握的： 第一，遵循先通用，再专用的原则。先设计好层次结构中每一级别的类中的通用部分，也就是提供给派生类继承的成员和标识为Public的成员；第二，在定义数据类型和存储区时要有预留量，以避免以后更改困难。例如，即使当前数据可能仅需要Integer类型就够了，在设计时我们使用 Long 类型的变量。当然，最好能物尽其用，也不要盲目放大；第三，在一个项目中，必须统一管理与分配团队中使用的所有的命名，以减少命名冲突，这一点其实事关重大；第四，要使用提供可行的最低访问权限的访问修饰符声明类成员。内部类成员应声明为 Private；仅在类内与派生类才需要的成员应标记为Protected；Friend 数据成员可以从类

的外部访问，但仅限于该模块是定义该类的项目的一个组成部分；使用Public标识的成员，只能是实例化时真正需要的内容，而且经常用在类层次结构的底部。”“ 也就是说，一个规范的操作，标准的命名体系可以决定基类的强壮与否？”我不禁感触了一声。“ 不对，应该这样说，可以决定的是给脆弱的基类穿上多厚的防护衣。因为基类始终都是脆弱的。”大李更正道。 我连声赞同：“ 对，对。我现在是真正明白为什么总有人提编程规范的事情，我一直认为是增强代码的可读性，没想到，对程序自身还有这么大的帮助。”“ 当然，其实你认真想一下，Overrides关键字的作用，不管要不要注明，编译器都可以很方便地判断方法或属性是否在基类中，签名是否匹配，但是VB.NET要求我们必须标注，就是强制开发人员注明重载基类方法或属性的意图，使开发过程更合理与有效。此外，还有更重要的就是，我们要在工程实践中不断地学习与磨练，了解更多的知识，获得更多的经验，这样才会成长为一名合格的程序设计师。就拿继承来说吧，在.NET中其实支持三种继承方式：实现继承、接口继承、可视继承。我们其实只用了第一种继承方式，你看，要学的东西是不是很多？”大李友好地拍了拍我的肩膀。 编辑特别推荐: 全国计算机等级考试二级VB模拟试题及答案解析 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