计算机二级VB辅导:构造与析构计算机等级考试 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_645172.htm 在Form1窗体的实例被隐藏的时候，关闭了Form2窗体的实例，使我失去了对主线程的人工控制，进程无法正常关闭了。只好使用Ctrl Alt Del调出系统进程管理器，强行中止了该进程。为了避免抬头看见大李的笑脸，我只好低头想办法。 有了，我只要能截获Form2实例关闭的消息，不就可以再调出隐藏的主线程窗体了吗？在Form2的基类事件(Base Class Event)中重载Closing方法进行处理： Private Sub Form2_Closing(ByVal sender As Object, ByVal e As _ System.ComponentModel.CancelEventArgs) Handles MyBase.Closing frm1.Show() End Sub 哈，很方便，我关闭了Form2窗体的实例后，被隐藏的那个frm1又出现了。 “ 嗯， ”大李终于发话了， “ 你再点击一下Form1窗体上的Button1试试。 ” “ 你关闭了frm2这个Form2的实例，也就结束了这个对象的生存期， ”看来是蓄势已久了， “ 这就是出错提示中所说的 ‘ 无法访问名为Form2的已处置对象 ’ 。当我们关闭一个窗口的时候，会发出一个终止响应，并将该窗口对象，就象上面定义的frm2，送入终止队列，公共语言运行库的垃圾回收器跟踪着这个对象的生存期，此时就会调用此对象基类，比如Form2的Dispose方法，用于销毁对象并收回资源。所以…… ” “ 所以我只要判断一下frm2是否被销毁就行了，如果销毁了，我就再构造一个实例不就行了？ ”我恍然大悟道。 “ 用frm2.IsDisposed就可以来判断了。 ”我心领神会地写道： Private Sub Button1_Click(ByVal sender As

System.Object, _ ByVal e As System.EventArgs) Handles Button1.Click If frm2 Is Nothing Or frm2.IsDisposed Then ‘ 判断对象是否被销毁 frm2 = New Form2() End If Me.Hide() frm2.Show() End Sub 这下完善多了，两个窗体之间的切换也不会有这么多别扭的问题了。我转过身，看到大李已经找了把椅子坐在我的身边。“ 你来说说，对VB.NET的窗体实例的创建与销毁的过程吧。 ”我整理了一下凌乱的思路，长吁了一口气，开始说：“ 一个窗体类，比如Form1类是通过调用其基类，就是Form类的New方法来创建实例、Dispose方法来销毁实例。 ”“ 没错。 ”大李边说话，一边在我的程序中点击开来被代码窗口自动折叠起来的" Windows 窗体设计器生成的代码 "： Public Sub New() MyBase.New() ’该调用是 Windows 窗体设计器所必需的。 InitializeComponent() ’在 InitializeComponent() 调用之后添加任何初始化 End Sub ’窗体重写处置以清理组件列表。 Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean) If disposing Then If Not (components Is Nothing) Then components.Dispose() End If End If MyBase.Dispose(disposing) End Sub 大李开始解说道：“ MyBase 关键字的行为类似于引用类的当前实例的基类的对象变量。MyBase 常用于访问在派生类中被重写或隐藏的基类成员。在这段代码中，MyBase指的当然就是System.Windows.Forms.Form类了。构造对象时用的New方法是显式调用的，没什么好解说的。但析构的方法值得一说。 ”他看了我一想，继续说：“ Form.Dispose方法是重写自Control.Dispose方法的，那么Control.Dispose方法的含义又是怎么样的？它的作用就是：释放由Control占用的非托管资

源，还可以另外再释放托管资源。当它参数中的disposing 为true 则释放托管资源和非托管资源；为 false 则仅释放非托管资源。 Form类的disposing为true。在关闭窗体时自动调用dispose的功能是得益于.net的公共语言运行库，运行库自动处理对象布局和管理对对象的引用，当不再使用对象时释放它们。其生存期以这种方式来管理的对象称为托管数据。自动内存管理消除了内存泄漏以及其他一些常见的编程错误。任何类型的 Dispose 方法都应该释放它拥有的所有资源。它还应该通过调用其父类型的 Dispose 方法释放其基类型拥有的所有资源。该父类型的 Dispose 方法应该释放它拥有的所有资源并同样也调用其父类型的 Dispose 方法，从而在整个基类型层次结构中传播该模式。要确保始终正确地清理资源，Dispose 方法应该可以被多次安全调用而不引发任何异常。 ”“ 可是，如果系统问题或应用程序调用上出了问题，不能正常调用Dispose怎么办？ ”我想起了什么，问道。“ 如果通过Dispose还释放不干净或没有调用Dispose，系统的垃圾回收器会调用对象的 Finalize 方法进行清除。由于执行 Finalize 方法会大大减损性能，所以我们不会一开始就用它去进行清除工作。 ”大李稍微解释了一下。 我终于想起了一个重要的问题：“ 如果总是在模块中定义的全局变量来处理，由于访问范围太大，会不会有安全性的问题？ ”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