C语言（上）

王继纲 编

目　　录

第一章.NET 编程语言 C#

Microsoft.NET 的概念.. .NET 框架.. C#语言在.NET 框架中的作用及其特性 1.1 Microsoft.NET 一场新的革命。

1.1.1 什么是.NET 2000 年 6 月 22 日不论对 Microsoft 还是对整个 IT 业界都将成为值得纪念的一天这一天微软公司正式推出了其下一代计算计划 Microsoft.NET(以下简称.NET) 这项计划将使微软现有的软件在 Web 时代不仅适用于传统的 PC 而且也能够满足目前呈强劲增长势头的新设备诸如蜂窝电话以及个人数字助理 Personal Digital Assistant, PDA 等的需要微软还计划通过创建新的工具来吸引软件开发人员和合作伙伴对 Microsoft.NET 的认同并且开发出其他基于 Internet 的服务那么你是否想知道究竟什么是.NET? 请听听微软官员的声音因特网的革命从微软的角度来讲我们就是要建设一个平台来创建并且支持新一代的应用 我们必须有一套通用系统服务来支持这样的操作这种观点就说明我们还有下一个层次的发展也就是说因特网下一步的发展它将使因特网的作用远远超越展现一个网站.NET 首先是一个开发平台它定义了一种公用语言子集 Common Language Subset CLS ,这是一种为符合其规范的语言与类库之间提供无缝集成的混合语.NET 统一了编程类库提供了对下一代网络通信标准可扩展标记语言 Extensible MarkupLanguage XML 的完全支持使应用程序的开发变得更容易更简单 Microsoft.NET 计划还将实现人

机交互方面的革命微软将在其软件中添加手写和语音识别的功能让人们能够与计算机进行更好的交流并在此基础上继续扩展功能增加对各种用户终端的支持能力最为重要的.NET 将改变因特网的行为方式软件将变成为服务与 Microsoft 的其它产品一样.NET 与 Windows 平台紧密集成并且与其它微软产品相比它更进一步由于其运行库已经与操作系统融合在了一起从广义上把它称为一个运行库也不为过简而言之.NET 是一种面向网络支持各种用户终端的开发平台环境微软的宏伟目标是让 Microsoft.NET 彻底改变软件的开发方式发行方式使用方式等等并且不止是针对微软一家而是面向所有开发商与运营商.NET 的核心内容之一就是要搭建第三代因特网平台这个网络平台将解决网站之间的协同合作问题从而最大限度地获取信息在 .NET 平台上不同网站之间通过相关的协定联系在一起网站之间形成自动交流协同工作提供最全面的服务。

1.1.2 我们为什么需要.NET 某一天你出差到外地在机场租借手机电话在向该终端插入自己的 IC 卡后自己的地址簿和计划簿被自动下载随即它就变成了你个人专用的 PDA 这不是梦境这是.NET 为我们描绘的一个未来生活的场景人们的需要总是无法满足我们不断地问自己我们还应该有些什么需求推动着技术的进步在二十一世纪 Internet 将成为商业活动的主要场所 B2B B2C 等电子商务的运作方式一对一营销的经营概念将网络的服务功能提高到了前所未有的程度微软公司在此时提出.NET 有其深远的战略考虑改革商务模型微软公司感觉到只靠销售软件包的商务模型没有什么前途该公司打算今后将中心转移到可以在网络上使用“服务”型商务这

样首要的问题就是解决网络上用来开发并执行“服务”的平台这就是 Microsoft.NET 提高软件开发生产效率并且试图使应用软件的发布更为容易再也不想因为 DLL 版本不同而烦恼希望不用重新启动电脑就能够安装应用软件改进用户界面并能支持多种用户终端用户界面演进的结果包括两方面的内容一是完成传统的 PC 界面与基于 XML 的浏览器界面间的过渡二是对自然语言和语音识别的支持从而使用户与各种终端之间的沟通更加透明真正达到网络互连的 3A Anywhere Anytime Any device 今天许多的人时常问除了上网看新闻我们究竟还能干什么这是因为今天的互联网与旧式的大型计算机的工作模式还有许多相似之处信息被储存在中央服务器内而用户的所有操作都要依靠它们让不同的网址之间相互传递有意义的信息或者合作提供更广泛和更深层次的服务还是一件十分困难的事现代人时常有一种困惑感觉到如今生活在技术与机器架构的丛林中我们在努力地去适应机器适应技术而不是机器和技术适应人类科技以人为本还只是一个美好的愿望这是因为我们还不能将控制信息的权利交给那些需要信息的人们 .NET 的出现意味着人们可以只用一种简单的界面就可以编写浏览编辑和分享信息而且还可以得到功能强大的信息管理工具由于使用的所有的文件都以符合网络协议的格式存在所以所有的商业用户和个人用户都可以方便地查找和使用其中的信息任何规模的公司都可以使用相同的工具与他们的供应商商业伙伴和客户高效地沟通和分享信息这样就创造出一种全新的协同工作模式总之.NET 战略是一场软件革命.. .NET 对最终用户来说非常重要因为计算机的功能将会得到大幅度提升同时计算机操作也会变得非常简单特

别地用户将完全摆脱人为的硬件束缚用户可以自由冲浪于因特网的多维时空自由访问自由查看自由使用自己的数据而不是束缚在便携式电脑的方寸空间——可通过任何桌面系统任何便携式电脑任何移动电话或 PDA 进行访问并可对其进行跨应用程序的集成.. .NET 对开发人员来说也十分重要因为它不但会改变开发人员开发应用程序的方式而且使得开发人员能创建出全新的各种应用程序大幅提高软件生产率.NET 将保证完全消除当今计算技术中的所有缺陷.NET 定能实现确保用户从任何地点任何设备都可访问其个人数据和应用程序的宏伟蓝图.. .NET 把雇员客户和商务应用程序整和成一个协调的能进行智能交互的整体而各公司无疑将是这场效率和生产力革命的最大受益者.NET 承诺为人类创造一个消除任何鸿沟的商务世界。

1.1.3 .NET 的核心组件.NET 的核心组件包括.. 一组用于创建互联网操作系统的构建块其中包括 Passport. NET 用于用户认证以及用于文件存储的服务用户首选项管理日历管理以及众多的其它任务.. 构建和管理新一代服务的基本结构和工具包括 Visual Studio.NET .NET 企业服务器.Net Framework 和 Windows.NET .. 能够启用新型智能互联网设备的.NET 设备软件.. .NET 用户体验 1.2 .NET 与 C# 1.2.1 支持多种编程语言的. NET 结构框架让我们翻开教科书回顾一下近十年来软件开发的历史多年以前当微软的组件对象模型 Compone nt Object Model, COM 尚未推出时软件的复用性对于开发人员仅仅是一种美好的憧憬成千上万的程序员为了处理通信接口和不同语言间的冲突而通宵达旦地艰辛劳动但却收效甚微 COM 的出现改变了这一切通过将组件

改变为通用集成型的构件开发人员正逐渐地从过去的繁复编程事务中解脱出来可以选择自己最得心应手的编程语言进行编程然而软件组件与应用程序之间的联合仍然是松散的不同的编程语言与开发平台限制了部件间的互用性其结果是产生了日益庞大的应用程序与不断升级的软硬件系统举个很简单的例子只用五行 C 语言代码就能编写出的一个简单程序若使用 COM 来编写结果会是令人吃惊的我们需要几百行代码 COM 在带来巨大价值的同时也大大增加了开发开销而.NET Framework 的出现使得一切问题都迎刃而解实际上在.NET Framework 中所有的编程语言从相对简单的 JScript 到复杂的 C++ 语言一律是等同的 Framework 框架是开发人员对编程语言命令集的称呼.Net 框架的意义就在于只用统一的命令集支持任何的编程语言正如微软 Web 服务中心的成组产品经理 John Montgomery 所说只需简单地一用.NET 框架便可消除各种异类框架之间的差异将它们合并为一个整体.NET 的作用不仅仅是将开发人员从必须掌握多种框架的束缚中解脱出来通过创建跨编程语言的公共 API 集.NET 框架可提供强大的跨语言继承性错误处理和调试功能现在开发人员可以自由地选择他们喜欢的编程语言.NET 平台欢迎所有人的垂顾 ”.NET 将使编程人员梦想的语言互用性变成为近在眼前的现实想想看一个在 Visual Basic VB 中定义的类能够在另一种与它完全不同的语言环境中使用调试甚至继承这是多么令人兴奋的事情.NET 框架是.NET 平台的基础架构其强大功能来自于公共语言运行时 Common Language Runtime,CLR 将在第二章中进行详细的解释环境和类库 CLR 和类库包括 Windows Forms ADO.NET 和 ASP.NET 紧

密结合在一起提供了不同系统之间交叉与综合的解决方案和服务.NET 框架创造了一个完全可操控的安全的和特性丰富的应用执行环境这不但使得应用程序的开发与发布更加简单并且成就了众多种类语言间的无缝集成。

1.2.2 面向.Net 的全新开发工具 C# 在最近的一段时间里 C 和 C++一直是最有生命力的程序设计语言这两种语言为程序员提供了丰富的功能高度的灵活性和强大的底层控制能力而这一切都不得不在效率上作出不同程度的牺牲如果你使用过包括 C 和 C++在内的多种程序设计语言相信你会深刻体会到它们之间的区别比如与 Visual Basic 相比 Visual C++程序员为实现同样的功能就要花费更长的开发周期由于 C 和 C++即为我们带来了高度的灵活性又使我们必须要忍受学习的艰苦和开发的长期性许多 C 和 C++程序员一直在寻求一种新的语言以图在开发能力和效率之间取得更好的平衡今天人们改进开发出了许多语言以提高软件生产率但这些或多或少都以牺牲 C 和 C++程序员所需要的灵活性为代价这样的解决方案在程序员身上套上了太多的枷锁限制了他们能力的发挥它们不能很好地与原有的系统兼容更为令人头痛的是它们并不总是与当前的 Web 应用结合得很好理想的解决方案是将快速的应用开发与对底层平台所有功能的访问紧密结合在一起程序员们需要一种环境它与 Web 标准完全同步并且具备与现存应用间方便地进行集成的能力除此之外程序员们喜欢它允许自己在需要时使用底层代码针对该问题微软的解决方案是一种称之为 C#的程序语言 C#是一种现代的面向对象的程序开发语言它使得程序员能够在新的微软.NET 平台上快速开发种类丰富的应用程序.NET 平台提供了大量的工具和

服务能够最大限度地发掘和使用计算及通信能力由于其一流的面向对象的设计从构建组件形式的高层商业对象到构造系统级应用程序你都会发现 C#将是最合适的选择使用 C#语言设计的组件能够用于 Web 服务这样通过 Internet 可以被运行于任何操作系统上任何编程语言所调用不但如此 C#还能为 C++程序员提供快捷的开发方式又没有丢掉 C 和 C++的基本特征强大的控制能力 C#与 C 和 C++有着很大程度上的相似性熟悉 C 和 C++的开发人员很快就能精通 C# 。

1.3 C#语言的特点 C#在带来对应用程序的快速开发能力的同时并没有牺牲 C 与 C++程序员所关心的各种特性它忠实地继承了 C 和 C++的优点如果你对 C 或 C++有所了解你会发现它是那样的熟悉即使你是一位新手 C#也不会给你带来任何其它的麻烦快速应用程序开发 Rapid Application Development RAD 的思想与简洁的语法将会使你迅速成为一名熟练的开发人员正如前文所述 C#是专门为.NET 应用而开发出的语言这从根本上保证了 C# 与.NET 框架的完美结合在.NET 运行库的支持下.NET 框架的各种优点在 C#中表现得淋漓尽致让我们先来看看 C#的一些突出的特点相信在以后的学习过程中你将会深深体会到# SHARP 的真正含义.. 简洁的语法.. 精心地面向对象设计.. 与 Web 的紧密结合.. 完整的安全性与错误处理.. 版本处理技术.. 灵活性与兼容性。

1.3.1 简洁的语法请原谅虽然我们一再强调学习本书不需要任何的编程基础但在这里还不得不提到 C++在缺省的情况下 C#的代码在.NET 框架提供的可操控环境下运行不允许直接地内存操作它所带来的最大特色是

没有了指针与此相关的那些在 C++中被疯狂使用的操作符例如-> 和.，已经不再出现 C#只支持一个. 对于我们来说现在需要理解的一切仅仅是名字嵌套而已 C#用真正的关键字换掉了那些把活动模板库 Active Template Library ALT 和 COM 搞得乱糟糟的伪关键字，如 OLE _COLOR BOOL VARIANT_BOOL DISPID_XXXXX 等等每种 C#类型在.NET 类库中都有了新名字语法中的冗余是 C++中的常见的问题比如 const ”和#define 各种各样的字符类型等等 C#对此进行了简化只保留了常见的形式而别的冗余形式从它的语法结构中被清除了出去。

1.3.2 精心地面向对象设计也许你会说从 Smalltalk 开始面向对象的话题就始终缠绕着任何一种现代程序设计语言的确 C#具有面向对象的语言所应有的一切特性封装继承与多态这并不出奇然而通过精心地面向对象设计从高级商业对象到系统级应用 C#是建造广泛组件的绝对选择在 C#的类型系统中每种类型都可以看作一个对象 C#提供了一个叫做装箱 boxing 与拆箱 unboxing 的机制来完成这种操作而不给使用者带来麻烦这在以后的章节中将进行更为详细的介绍 C#只允许单继承即一个类不会有多个基类从而避免了类型定义的混乱在后面的学习中你很快会发现 C#中没有了全局函数没有了全局变量也没有了全局常数一切的一切都必须封装在一个类之中你的代码将具有更好的可读性并且减少了发生命名冲突的可能整个 C#的类模型是建立在.NET 虚拟对象系统 Visual Object System VOS 的基础之上其对象模型是.NET 基础架构的一部分而不再是其本身的组成成分在下面将会谈到这样做的另一个好处是兼容性借助

于从 VB 中得来的丰富的 RAD 经验 C#具备了良好的开发环境结合自身强大的面向对象功能 C#使得开发人员的生产效率得到极大的提高对于公司而言软件开发周期的缩短将能使它们更好地应付网络经济的竞争在功能与效率的杠杆上人们终于找到了支点。

1.3.3 与 Web 的紧密结合.NET 中新的应用程序开发模型意味着越来越多的解决方案需要与 Web 标准相统一例如超文本标记语言 Hypertext Markup Language HTML 和 XML 由于历史的原因现存的一些开发工具不能与 Web 紧密地结合 SOAP 的使用使得 C#克服了这一缺陷大规模深层次的分布式开发从此成为可能由于有了 Web 服务框架的帮助对程序员来说网络服务看起来就像是 C#的本地对象程序员们能够利用他们已有的面向对象的知识与技巧开发 Web 服务仅需要使用简单的 C#语言结构 C#组件将能够方便地为 Web 服务并允许它们通过 Internet 被运行在任何操作系统上的任何语言所调用举个例子 XML 已经成为网络中数据结构传送的标准为了提高效率 C#允许直接将 XML 数据映射成为结构这样就可以有效地处理各种数据。

1.3.4 完全的安全性与错误处理语言的安全性与错误处理能力是衡量一种语言是否优秀的重要依据任何人都会犯错误即使是最熟练的程序员也不例外忘记变量的初始化对不属于自己管理范围的内存空间进行修改这些错误常常产生难以预见的后果一旦这样的软件被投入使用寻找与改正这些简单错误的代价将会是让人无法承受的 C#的先进设计思想可以消除软件开发中的许多常见错误并提供了包括类型安全在内的完整的安全性能为了减少开发中的错误 C#会帮助开发者通过更少的代码完

成相同的功能这不但减轻了编程人员的工作量同时更有效地避免了错误发生.NET 运行库提供了代码访问安全特性它允许管理员和用户根据代码的 ID 来配置安全等级在缺省情况下从 Internet 和 Intranet 下载的代码都不允许访问任何本地文件和资源比方说一个在网络上的共享目录中运行的程序如果它要访问本地的一些资源那么异常将被触发它将会无情地被异常扔出去若拷贝到本地硬盘上运行则一切正常内存管理中的垃圾收集机制减轻了开发人员对内存管理的负担.NET 平台提供的垃圾收集器 Garbage Collection GC 将负责资源的释放与对象撤销时的内存清理工作变量是类型安全的 C#中不能使用未初始化的变量对象的成员变量由编译器负责将其置为零当局部变量未经初始化而被使用时编译器将做出提醒 C#不支持不安全的指向不能将整数指向引用类型例如对象当进行下行指向时 C#将自动验证指向的有效性 C#中提供了边界检查与溢出检查功能。

1.3.5 版本处理技术 C#提供内置的版本支持来减少开发费用使用 C#将会使开发人员更加轻易地开发和维护各种商业应用升级软件系统中的组件模块是一件容易产生错误的工作在代码修改过程中可能对现存的软件产生影响很有可能导致程序的崩溃为了帮助开发人员处理这些问题 C#在语言中内置了版本控制功能例如函数重载必须被显式地声明而不会像在 C++或 Java 中经常发生的那样不经意地被进行这可以防止代码级错误和保留版本化的特性另一个相关的特性是接口和接口继承的支持这些特性可以保证复杂的软件可以被方便地开发和升级。

1.3.6 灵活性和兼容性在简化语法的同时 C#并没有

失去灵活性尽管它不是一种无限制的语言比如它不能用来开发硬件驱动程序在默认的状态下没有指针等等但是在学习过程中你将发现它仍然是那样的灵巧如果需要 C#允许你将某些类或者类的某些方法声明为非安全的这样一来你将能够使用指针结构和静态数组并且调用这些非安全的代码不会带来任何其它的问题此外它还提供了一个另外的东西这样的称呼多少有些不敬来模拟指针的功能 delegates 代表再举一个例子 C#不支持类的多继承但是通过对接口的继承你将获得这一功能下面谈谈兼容性正是由于其灵活性 C#允许与 C 风格的需要传递指针型参数的 API 进行交互操作 DLL 的任何入口点都可以在程序中进行访问 C#遵守.NET 公用语言规范 Common Language Specification CLS 从而保证了 C#组件与其它语言组件间的互操作性元数据 Metadata 概念的引入既保证了兼容性又实现了类型安全。

1.4 小 结 Microsoft.NET 计划将彻底改变我们对因特网的认识从而在这样一个网络时代彻底改变我们的生活软件是一种服务技术是我们的仆人时间与地点将不再是我们面前的障碍建立在 CLR 与类库基础上的.NET 框架是.NET 平台的核心组件之一这为软件的可移植性与可扩展能力奠定了坚实的基础并为 C#语言的应用创造了良好的环境 C#是.NET 平台的通用开发工具它能够建造所有的.NET 应用其固有的特性保证了它是一种高效安全灵活的现代程序设计语言从最普通的应用到大规模的商业开发 C#与.NET 平台的结合将为你提供完整的解决方案在本章中我们提出了与.NET 以及与 C#语言相关的一些概念例如 CLR VOS 和 GC 也许你是初次接触它们但不用担心在以后的各章中我们将详细地介绍

这些相关的概念与知识相信通过学习你将能够迅速掌握它们并熟练地运用它们提供的各种特性。

第二章运行环境全面了解.NET

C#运行在.NET 平台之上其各种特性与.NET 密切联系它没有自己的运行库许多强大的功能均来自.NET 平台的支持因此要想真正掌握 C#首先必须了解.NET 本章将向你介绍 C#的运行环境重点放在.NET 公用语言运行时环境与公用语言规范上最后介绍了.NET 的开发工具。

2.1 .NET 结构.NET 包括四个组成部分.. VOS 类型系统.. 元数据.. 公用语言规范.. 虚拟执行系统下面分别对它们进行简要介绍。

2.1.1 虚拟对象系统.NET 跨语言集成的特性来自于虚拟对象系统 VOS 的支持在不同语言间进行代码复用和应用集成中所遇到的最大问题是不同语言类型系统间的相容性问题可以想象不同的语言虽然语法结构大体相同但数据类型与语言环境本身的各种特点联系紧密很难想象一种解释性的语言所拥有的数据类型会与一种编译语言相同而即使相同的数据类型在不同的语言环境中表示的意义也存在差别例如同样是整数类型在 MSSQL 中的长度是 32 位而在 VB 中却是 16 位至于日期时间与字符串类型在这方面的区别就更加明显了 VOS 的建立就是为了改变这种状况它既支持过程性语言也支持面向对象的语言同时提供了一个类型丰富的系统来容纳它所支持的各种语言的特性它在最大程度上屏蔽了不同语

言类型系统间的转换使程序员能够随心所欲地选择自己喜欢的语言当然这种语言必须支持.NET 应用从事开发保证了不同语言间的集成对于过程性语言它描述了值的类型并指定了类型的所有值必须遵守的规则在面向对象的语言方面它统一了不同编程语言的对象模型每一个对象在 VOS 中都被唯一标识以与其它对象相区别。

2.1.2 元数据元数据是对 VOS 中类型描述代码的一种称呼在编译程序将源代码转换成为中间代码时它将自动生成并与编译后的源代码共同包含在二进制代码文件中元数据携带了源代码中类型信息的描述这在一定程度上解决了版本问题程序使用的类型描述与其自身绑定在一起在 CLR 定位与装载类型时系统通过读取并解析元数据来获得应用程序中的类型信息 JIT 编译器获得加载的类型信息后将中间语言代码翻译成为本地代码在此基础上根据程序或用户要求建立类型的实例由于整个过程中 CLR 始终根据元数据建立并管理对应特定应用程序的类型从而保证了类型安全性此外元数据在解决方法的调用建立运行期上下文界限等方面都有着自己的作用而关于元数据的一切都由.NET 在后台完成。

2.1.3 公用语言规范公用语言规范 Common Language Specification CLS 是 CLR 定义的语言特性集合主要用来解决互操作问题如果一个类库遵守 CLS 那么同样遵守 CLS 规范的其它编程语言将能够使用它的外部可见项详细的内容见本章第二节 2.1.4 虚拟执行系统虚拟执行系统 Visual Execution System VES 是 VOS 的实现它用来驱动运行环境元数据的生成与使用公用语言规范的满足性检查以及应用程序执行过程中的内存管理均由它来完成具体说来 VES 主要完成以下功能.. 装入

中间代码.. 使用 JIT 将中间代码转换为本地码.. 装入元数据.. 代码管理服务包括垃圾收集器和异常处理.. 定制与调试服务.. 线程和环境管理。

2.2 公用语言运行时环境与公用语言规范了解了.NET 的结构之后我们该看看.NET 利用其结构为我们创造的运行环境公用语言运行时环境它是 C#及其它支持.NET 平台的开发工具的运行基础具体来说它为我们的应用提供了以下益处.. 跨语言集成的能力.. 跨语言异常处理.. 内存管理自动化.. 强化的安全措施.. 版本处理技术.. 组件交互的简化模型。

2.2.1 理解 CLR .NET 提供了一个运行时环境叫做公用语言运行时它管理着代码的执行并使得开发过程变得更加简单这是一种可操控的执行环境其功能通过编译器与其它工具共同展现你的代码将受益于这一环境依靠一种以运行时为目标的指完全支持运行时环境的编译器所开发的代码叫做可操控代码它得益于可操控环境的各种特性跨语言集成跨语言异常处理增强的安全性版本处理与开发支持简单的组件交互模型以及调试服务为了使运行时环境能够向可操控代码提供服务语言编译器需要产生一种元数据它将提供在你使用语言中的类型成员引用的信息元数据与代码一起存储每个可加载的 CLR 映像均包含了元数据运行时环境使用元数据定位并载入类在内存中展开对象实例解决方法调用产生本地代码强制执行安全性并建立运行时环境的边界运行时环境自动处理对象的展开与引用当它们不再使用时负责它们的释放被运行时环境进行这样的生命期管理的对象被称为可操控代码自动内存管理消除了内存溢出同时也解决了其它一些常见的语法错误如果你的代码是可操控的你仍然可

以在需要的时候使用非可控代码或者在你的.NET 应用中同时使用可控与非可控代码由于语言编译器支持他们自己的类型比如一些原始类型你可能并不总是知道也不必知道你的数据是否是可控的 CLR 使设计跨语言的组件与应用变得更加容易以不同语言设计的对象能够彼此间进行通信并且它们的行为能够紧密地综合与协调举个例子你定义了一个类然后可以在另一种不同的语言中从该类中派生了一个类或者调用它其中的一个方法你也可以向另一种语言中类的方法传递该类的一个实例这种跨语言的集成之所以可能因为以运行时间为目标的语言编译器与工具使用一种运行时间所定义的公用类型系统他们遵守运行时的规则公用语言规范来定义新的类型生成使用保持并绑定类型作为元数据的一部分所有可控组件携带了关于它们所依赖的组件与资源的信息运行时环境使用这些信息来保证你的组件或应用具有需要的所有东西的特定版本其结果是你的代码将不会因为版本冲突而崩溃注册信息与状态数据不再保存在难以建立与维护的注册表中你所定义的类型及附属信息作为元数据被保存这使得复制与移动组件的复杂程度得到降低编译工具用他们自己的方式向开发人员展现 CLR 的功能这意味着运行时间的一些特性可能在不同的语言中的表现形式将会有所不同你怎样体验运行时的特性将取决于你所使用的语言比如说如果你是一位 VB 开发人员你可能注意到在运行时环境的帮助下 VB 语言比以前具有更多的面向对象的特性。

2.2.2 可操控执行的含义前面的叙述中我们多次提到了可操控这一概念这意味着它指向的对象在执行过程中完全被运行时环境所控制在执行过程中运行时环境提