



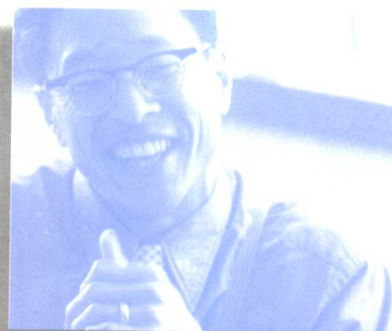
微软培训教材系列

Designed for
Microsoft®
Windows NT®



CD-ROM
Included

Visual
++
开发教程



Microsoft® **Visual J++™** 开发教程

北京博彦科技发展有限公司 编译



清华大学出版社
<http://www.tup.tsinghua.edu.cn>

微软培训教材系列

Visual J++ 开发教程

北京博彦科技发展有限责任公司 编译

陈、
博彦

清华大学出版社

(京)新登字 158 号

内 容 提 要

Visual J++ 是 Microsoft 为 Java 语言开发的 Windows 程序集成开发环境,功能强大,表达能力强,但对于初学者有一定的难度。本书将一步步由浅入深介绍 Visual J++ 的环境和使用、类与窗体、菜单、方法、继承、创建小应用程序、保存信息、Java 中的软件包、控制台应用程序、使用 MFC 中的组件、可移植性以及访问数据库等内容。对于希望学习 Visual J++ 的各类人员,即使没有 Windows 环境的编程经验,也可以使用本教程逐步进行学习。

书后实验从使用应用程序向导创建应用程序开始,首先熟悉开发环境、为简单的程序添加丰富的内容,然后介绍创建小程序等更深的内容,覆盖了教程中涉及的全部内容,具有一定的连贯性,希望读者能够按照练习的顺序依次进行操作,以免对练习中涉及的某些内容感到不解。练习中创建程序的源代码和支持练习的数据库文件可以在随书光盘的 Lab 文件夹中得到。虽然本书中的练习可以单独完成,但是建议在教师的指导下进行,以便在出现与所提供结果不同时,可以从指导教师处得到相应的解释。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: Visual J++ 开发教程

作 者: 北京博彦科技发展有限公司 编译

出 版 者: 清华大学出版社(北京清华大学校内,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 胡先福

印 刷 者: 清华大学印刷厂印刷

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印张:** 17.75 **字数:** 418 千字

版 次: 2000 年 9 月第 1 版 2000 年 9 月第 1 次印刷

书 号: ISBN 7-900630-14-7

印 数: 0001~4000

定 价: 39.00 元

《微软培训教材系列》序

目前,大学计算机教育远远落后于计算机的实际发展,计算机教程只注重基础知识和原理的学习,而忽略了计算机新技术、新知识的学习,结果导致学生对计算机新技术、新知识的认识和掌握明显不足,造成学生的知识结构和社会实际需求的严重脱节。

基于以上状况,经过清华大学出版社、微软中国公司和北京博彦科技发展有限公司讨论决定,由三方合作编写、出版微软计算机技术培训系列教程。本套教程是微软计算机最新技术循序渐进的指导书。使用本套教程您可以快速便捷地掌握最新微软计算机技术,并得心应手地解决实际问题。

主要内容

本套教程由微软出版社提供相应的资料,授权博彦公司改编,根据大学的选修课程和各类计算机技能培训的要求,介绍微软公司最新的操作系统、办公软件、开发平台以及网络应用等各领域的操作方法和实用技巧。本套教程的选题以微软最新的软件、技术为依据,教材与技术同步更新,以保证内容的先进性。

读者范围

本套教程定位于初、中级用户,适用于大学的选修课程以及各类计算机技能培训。不管用户是否接触过本套教程介绍的内容,本套书都将对您非常适合。新的用户通过本套教程可以轻松达到中级用户水平;老用户可以从本套教程中了解到新版本软件的新特色、新功能,从而掌握新版本软件的使用技巧。

教程特色

在学习计算机技术过程中,如果没有机会直接去探究工作环境中的菜单、选项、操作步骤以及相应的反应,要理解和使用计算机技术是十分困难的。为此本套教程提供了一套相对简单但是必不可少的实验练习,来测试教程中介绍的内容,以达到理论和实践相结合的目的。

虽然本套教程中的练习可以单独完成,但是建议最好有教师在一旁进行指导,以便在出现不同的情况时,可以从指导教师处得到相应的解释。

教程风格

本套教程力求语言精练、图文并茂。对本套教程没有涉及到的更深一层的内容给出了相应的提示,为读者指点进阶捷径。

俗话说:“到达卡耐基殿堂的途径是练习、练习、再练习。”这话对个人成功和学习计算机技术都是非常有道理的。学习计算机技术的最好的方法就是理论和实践紧密结合。本套教程无疑是您的最佳选择。

经过数月的艰苦工作,这套教程终于可以奉献给广大读者了。这首先要感谢微软出版社为这套教程提供了大量资料;还要感谢清华大学出版社的各位编辑,正是他们一丝不苟的审校,才使这套教程得以顺利出版。

为用户提供最好的中文版软件和最好的计算机图书是我们努力的目标。如果您能从我们的软件和图书中获得有用的知识和技巧,那将是我们最大的欣慰。

由于时间仓促、水平有限,不当之处在所难免,恳请读者批评指正。

博彦公司

2000年4月

前 言

Visual J++ 是 Microsoft 为 Java 语言开发的用于 Windows 编程的集成开发环境。近几年来,越来越多的人开始学习 Visual J++ 语言。这主要是因为随着计算机网络的发展,Internet 风靡世界,通过 Web 浏览器可以在世界各地漫游、获取各种丰富的信息。而 Visual J++ 语言正是帮助我们方便快捷地开发基于 Web 的小程序和 Windows 环境下的应用程序的有力工具。

Java 语言功能强大,表达能力强,对于初学者有一定的难度,建议结合计算机编程环境进行学习,这样会达到事半功倍的效果。尽管并不需要有 Windows 环境的编程经验,但是要从本书中学到尽可能多的知识,应把自己视为 Windows 系统的用户。读者很快就会发现,在 Windows 环境下使用 Microsoft Visual Studio 和 Visual J++ 是非常高效便利的。

本教程将由浅入深地使读者熟悉 Java 语言和 Visual J++ 环境。其中第 1 章和第 2 章介绍 Java 语言及其基础,第 3 章介绍增强程序功能以及调试程序,第 4 章介绍类和窗体,第 5 章介绍菜单、方法和继承,第 6 章介绍如何创建小程序,第 7 章介绍如何保存信息,第 8 章介绍如何创建能显示动画的小程序,第 9 章介绍 Java 中的软件包,第 10 章介绍控制台应用程序,第 11 章介绍如何使用 MFC 的外部组件,第 12 章介绍 I/O 软件包,第 13 章介绍数据库技术。最后是本教程的实验部分,用于检验前面学习的内容。

本教程的书后附有教程的配套光盘,帮助您巩固教程中的内容。光盘中的内容包括教程和实验中使用的例子程序,在这些程序中可以得到完整的代码,以供参考。

此外,在学习本教程之前,应该具备一些编程语言的基本知识,包括程序流程、循环、分支、函数或子程序调用、参数传递以及编译程序等。

目 录

第 1 章	Java 概述	1
1.1	Java 语言概述	1
1.2	Visual J++ 开发环境	5
第 2 章	Java 语言基础	14
2.1	符号	14
2.2	数据类型	19
2.3	表达式	23
2.4	控制流程	23
第 3 章	增强 Hello 应用程序	31
3.1	向窗体中添加控件	31
3.2	应用程序的设计方法	35
3.3	增强应用程序的功能	36
3.4	插入注释	39
3.5	调试代码	42
第 4 章	类与窗体	49
4.1	类与对象	49
4.2	创建新的 Java 类	51
4.3	创建对象	52
4.4	命名规则	57
4.5	构造方法	57
4.6	添加对话框和附加窗体	60
第 5 章	菜单、方法和继承	71
5.1	成员变量修饰符	71
5.2	菜单操作	76
5.3	方法	79
5.4	继承	87
第 6 章	创建小程序	96
6.1	小程序	96
6.2	创建小程序	98
6.3	初识 HTML	100

6.4	从头创建小程序	105
6.5	java.awt 软件包	106
6.6	响应小程序事件	108
6.7	AWT 中的组件	114
6.8	在小程序中使用多媒体文件	125
6.9	给小程序传递参数	128
6.10	小程序和 Web 页	130
第 7 章	保存信息	140
7.1	使用数组	140
7.2	使用文件	146
7.3	使用列表	154
7.4	接口	156
7.5	数组和列表之间的转换	159
第 8 章	小程序中的动画	162
8.1	多线程	162
8.2	异常处理	167
8.3	编写动画代码	173
第 9 章	Java 中的软件包	177
9.1	软件包的概念	177
9.2	Java 软件包	180
9.3	WFC 软件包	181
9.4	WFC 应用程序软件包	182
9.5	创建自己的软件包	188
第 10 章	控制台应用程序	190
10.1	控制台应用程序的不同之处	190
10.2	控制台 I/O	195
第 11 章	使用 MFC 的外部组件	199
11.1	Win32 API	199
11.2	J/Direct Call Builder	200
11.3	利用附加控件工作	202
11.4	COM 组件	206
第 12 章	可移植 I/O	210
12.1	处理文件	210
12.2	访问 Internet	218

第 13 章 访问数据库	219
13.1 运行 Data form Wizard 创建应用程序.....	219
13.2 不使用数据库向导访问数据库	226
实验	233
实验 1	233
实验 2	235
实验 3	237
实验 4	238
实验 5	243
实验 6	251
实验 7	257
实验 8	259
实验 9	264
实验 10	266
实验 11	269

第 1 章 Java 概述

欢迎学习 Visual J++ !

在本章中,将介绍以下内容:

- Java 语言概述
- Microsoft Visual J++ 6.0 开发环境
- 第一个 Visual J++ 应用程序

1.1 Java 语言概述

Java 编程语言非常引人注目。其中一个主要的原因是因为 Java 语言是在 Internet 上编程最明智的选择。人们谈论使用 Java 语言时,涉及最多的就是这种语言与众不同的设计方法。本节介绍一些关于 Java 语言的基本信息。虽然这些信息对于学习 Java 语言编程并不是必不可少的,但这是极好的背景知识。

Java 首先是一种编程语言,它具有以下几方面的主要特点:

- 它是一种面向对象的编程语言
- 它的语法结构与 C 和 C++ 语言极为相近
- Java 所用的语法十分简单
- 具有良好的可移植性

下面我们深入看看每一种特征。

1.1.1 面向对象

我们知道,如果将一个大的计算机源程序作为整体来编写,工作会十分复杂和繁重。因此,人们往往将一个程序划分成若干小的模块,各个模块采用不同的方案,这样,就可以使各个模块之间保持相对的独立性,以便不同的人可以同时编写不同的模块。

程序员在维护程序时,需要对其作适当的修改,因此这些模块的组成结构显得极为重要。在运行维护过程中,人们往往希望程序可以实现更为理想的功能,或者希望利用操作系统的新特性,或者希望检验程序的某项功能。如果组成应用程序的各个模块之间不能保证相对独立的话,程序员在维护时可能要改动几个模块,而这些改动就很可能引起其他更多的模块要做出相应的改动,新的改动又会引起必要的相应改动,依此类推。如果这样的话,软件的维护工作简直无法进行。许多资料表明,面向对象编程语言(Object Oriented

Programming, OOP) 是解决应用程序的模块化及维护问题的有效技术。在面向对象语言编程中, 各个模块以应用程序处理的对象为基础, 就像用功能分解技术创建子模块一样。在功能分解中, 各个子模块又以应用程序本身提供的函数为基础。

下面, 我们来看一个示例: 存货清单控制应用程序。在 OOP 中, 对象包括: 存货清单条款、部件数、停放明细表、存储位置等。在功能分解过程中, 可以有接收、加标签、包装及运输等诸如此类的功能。如果部件数的格式发生变化, 比如说从 6 位数字变为 6 位字符, 那么, 对于一个优秀的 OOP 解决方案, 这些改动就应该可以在一个部件数模块中完成, 而不必对其他模块进行改动。当然, 其他模块对部件数模块会有调用, 只是在这些模块中, 把部件数模块作为一个“黑箱”实体。也就是说, 在这些调用部件数的模块当中, 不包含任何关于部件数模块的组建和安装信息。通常, 在非 OOP 的功能分解方案中, 像部件数那样的基本实体, 在应用程序中可能直接对每个模块操作, 也就是说, 如果部件数的格式发生了变动, 也就意味着所有引用部件数的模块都要更新和调整。所有这些都进一步说明了模块化设计的好处。任何良好的面向对象的程序, 都应比单单基于功能分解的方法更加容易编写和维护。

目前, 已有很多关于对象及 OOP 的书籍, 本书并没有对 OOP 进行详尽完备的讨论, 只是在以后的章节中对对象与编程知识进行了简单的介绍。

1.1.2 类似 C 和 C++ 的语法结构

编程语言的语法结构, 是指将语言的各个部分有机地组合起来而形成程序的方式。刚一接触, 你就会发现, Java 程序的确与 C 和 C++ 程序极为相似。C 语言是一种效率很高、灵活性很强的计算机编程语言。相当一部分商业软件是用 C 语言编写的, 然而, 它没有内建的 OOP 支持。C++ 的语法同 C 类似, 但它有了内建的 OOP 支持。C 和 C++ 都是非常流行的计算机编程语言, 目前的大部分软件都是由 C 或 C++ 编写的。由于 Java 语言的语法与 C 和 C++ 语言的语法类似, 因此, 可以很容易地将 C 或 C++ 程序改写成 Java 程序。

1.1.3 简化的语法结构

Java 语言的发明者拥有丰富的 C 和 C++ 语言的经验, 他们在吸取 C++ 优秀特征的同时, 舍弃了 C++ 中一些容易引起问题但不是绝对必要的部分(遗憾的是, 他们并没有完全解决这些问题)。

C 和 C++ 语言中都有指针。我们知道, 指针是 C 和 C++ 语言完成一些高级程序设计的基础, 然而, 也正是指针容易引起问题, 有时甚至会带来致命的麻烦。可以认为, 指针是一把双刃剑, 如果能够正确使用, 它能实现许多功能, 而一旦使用失误, 就有可能给用户带来灾难。Java 语言取消了指针, 而使用引用。这样, 通过使用引用, 不但保留了指针的灵活性, 而且去除了许多由指针带来的潜在隐患。C 语言的基本组建模块是函数, 这极适合功能分解; Java 语言将类作为它的基本组建模块, 非常适于处理对象。因此, Java 语言适

于面向对象编程。这些转变去除了 C++ 语法中许多含糊不清的部分。

以上都说明了 Java 语言对语法作了大量的简化,它比 C 和 C++ 语言都要容易学习。

1.1.4 可移植性

良好的可移植性也是 Java 语言倍受青睐的一个原因。具有良好的可移植性,就可以灵活地从 Web 站点上下载并使用应用程序。

取得程序的传统方式

你所创建的程序,一般都是针对某一种专门的计算机和操作系统而设计的,程序只能在专门的机器和操作系统上运行。因此,在 PC 机上创建的程序只能在 PC 机上运行,而不能在其他机器(如运行 UNIX 操作系统的工作站或大型机)上运行。这样,用户只能到计算机商店去购买软件,而且只能购买自己计算机的专用软件。

取得程序的 Web 方式

在 Web 页上,就与上述情况不同了。用户可以从网络服务器上将程序下载到自己的计算机上。HTML、图像和声音等都可以下载。网络浏览器会将这些片段组织到 Web 页上,以供用户查看。当需要在网络上运行程序时,根本无法保证所有的网络用户都使用同样的计算机,这就要求在 Web 上运行的程序能够在任何计算机上运行,而 Java 语言恰恰是为了实现这一目的而设计的。

编译器

目前流行的计算机语言都采用英语,以便于程序员记忆和开发应用程序。但是,我们知道,一台计算机是无论如何也不会懂得诸如 if、public 或 return 的含义的,这就需要编译器将这些代码翻译成计算机能懂的机器代码。不同类型的计算机使用不同的机器代码,Java 语言旨在为不同的计算机设计应用程序。

Java 的字节代码和虚拟机

Java 编译器将 Java 源代码翻译成机器码。然而,这些机器码并不是特地为 PC 机、工作站或大型机而生成的,而是为一台并不存在的“虚拟机”(Virtual Machine, VM)生成的。也就是说,Java 语言为虚拟机生成机器码。这种虚拟机的 Java 机器码也称作字节码(byte-code)。那么,为一个不存在的机器生成机器码,究竟有什么好处呢? Java 的虚拟机实际上是一个程序,这个程序实现字节码向实际机器代码的转换。如果没有考虑到在 Web 上编程的难度,以上做法听起来似乎有一些奇怪。在网络上,可能会有很多用户访问你的网址,而谁也不可能知道这些用户正在或将要使用什么类型的计算机。使用传统的编程语言,你不得不为每一种可能的机器类型编写一种版本的应用程序,然后,还要确保这些程序能为对应机型的用户所下载。如果网络访问者并未告知他的机器类型,那就不得不应付层出不穷的对话框,例如“单击此处查看此页”、“如果你在西班牙,请单击此按钮”,等

等。显然,这对一般用户来说是难以容忍的。因此,应用了虚拟机的字节码也是 Java 语言又一个主要的魅力所在。

良好的移植性

Java 语言良好的移植性,使得它可以创建一些精巧的小程序,在 Web 上完成动画设计和人机交互的功能,但这也有些不利的地方。

第一,影响速度。使用其他编程语言,机器码在程序运行时已经就绪,而使用 Java 语言,首先要生成对应虚拟机的字节码,然后进一步生成真机代码。

第二,Java 语言依赖 Java 虚拟机。你一定会经常遇到在原来的操作系统上更新软件版本的问题。Java 虚拟机也有自己的版本,单是 Windows 环境下就有 12 个版本。同样的计算机,使用不同的虚拟机版本,其运行情况会有相当大的差别,有时运行非常良好,有时可能会无法运行。

第三,移植性本身就是有争议的。在一个系统上,可以安装有 1 个、2 个、3 个乃至 5 个鼠标键的鼠标(X Window System 支持 5 个鼠标键的鼠标器)。为了保证程序的可移植性,在设计一个程序时,就必须考虑支持单键鼠标,现在,大家都在使用双键鼠标,于是支持单键鼠标的 Java 程序就不免显得过于蹩脚。

第四,一些技术问题使得 Web 网络尚且不能成为人们传递信息的基本工具。其中一个最主要的原因是由于下载复杂软件要消耗大量时间。在有些情况下,Java 程序良好的移植性的吸引力被其巨大的尺寸所抹杀了。

第五,许多人往往只是为自己所在公司的小网络编写程序,而不是编写 Internet 软件,这些小网络有时称为企业网或内部网。一般来说,在一个公司的内部所用的计算机类型和计算机软件的版本都是标准化的,因而也就很少追求一种完全可移植的软件。内部网一般都有足够的带宽,可以保证下载信息的速度。

因此,程序的可移植性并不总是人们所最关注的。尽管如此,下载技术的发展仍为我们提供了令人激动的前景:自动更新。

自动更新

我们知道,每次访问 Web 站点时,都可以从某个网址得到新信息内容的副本。如果有人每天都将网址的内容更新一次,那么,访问者每天都可以得到新的内容。对于 Java 字节码也是如此,如果有人将新的 Java 字节码载入 Web 站点,以后就会有人访问相应的 Web 站点,将这些新的 Java 字节码下载到自己的计算机上。这是相当明智的做法,你可以毫不费力地将软件的维护与更新工作分配给网友,可以通过网络下载最新版本的应用软件。

Java 软件包

学习计算机编程的经验表明,应用程序所能完成的功能,要么是语言本身提供的,要么是我们自己编写的。例如 Fortran 语言,从磁盘或磁带等存储器读写数据的命令包含在 Fortran 语言的内部。大多数新的编程语言都带有一个拥有许多实用子程序的函数库。这

些子程序处理函数就同从存储器上读写数据类似。函数库具有以下两点好处：

- 库函数对于语言来说是外部的,使得语言更加简洁。
- 库可以独立于语言而不断地更新。

Java 语言也带有这样的库,称作 Java API。Java API 非常庞大,被分割为几个称作“包”的功能模块。包按功能可以分为绘图、处理存储和创建小程序等。事实上,几乎所有实现可移植性的代码都在 Java API 当中。

表 1-1 包含了一些较为重要的 Java API 包以及所支持的特性。

表 1-1 Java API 中经常使用的包

API 包	支持	API 包	支持
java.applet	通用 applet 特性	java.net	网络
java.awt	Windows 和 GUI 特性	java.security	安全
java.io	数据输入和输出	java.text	文字处理特性
java.lang	核心语言特性	java.util	各种实用工具特性
java.math	数学指令		

有些软件包与 Java 语言本身的联系相当紧密。例如,如果不使用 java.lang 软件包,就不能编写最简单的 Java 程序。java.lang 软件包包括 String 和 Object 类,如果不使用 String 类,编写 Java 程序就会十分困难;如果不使用 Object 类,就绝对不可能编写 Java 程序。其他软件包,例如 java.awt,它的用途非常专一。这个软件包处理图形用户界面(GUI)元素,比如显示窗口、菜单和按钮等。java.awt 用以支持 Java 语言 GUI 的可移植性。

WFC (Windows Foundation Class)库是 Visual J++ 提供的类库集,目的是为 Java 提供完整的、最吸引人的组件和编程模式。它其中包含的 com.ms.wfc.ui 软件包,可以代替 java.awt 实现窗口、菜单和按钮等控件的显示。WFC 的内容将在以后的章节当中进一步阐述。

外部库文件的使用使得程序员能够为自己的 GUI 软件包选择 java.awt 或者 WFC。

1.2 Visual J++ 开发环境

Microsoft Visual J++ 是利用 Java 语言创建 Java 应用程序的工具。Microsoft Visual J++ 是 Microsoft Visual Studio 套件中的一个成员,适合于创建应用程序。Microsoft Visual Studio 套件中的其他成员还包括 Microsoft Visual C++ 和 Microsoft Visual Basic。Microsoft Visual C++ 为创建强有力的应用程序提供了良好的开发环境,其中包括高级调试器、灵活的工程文件组建系统,以及高度优化的编译器。Microsoft Visual Basic 是一个快速的应用程序开发环境,Visual Basic 语言因为可以创建 Windows 应用程序而倍受青睐。Microsoft Visual J++ 继承了上述两种语言各自的优点。

Visual J++ 的早期版本拥有同 Visual C++ 相类似的开发环境,而 Visual J++ 的最新版引入了 Visual Basic 开发系统的许多特征,包括可视窗体设计和属性窗口。Visual J++

继承了许多使 Visual Basic 得以成功的特点,比如拥有强有力的编程环境,综合了高质量的应用程序设计、代码生成、组建和调试功能等。这便是集成开发环境(Integrated Development Environment, IDE)。

1.2.1 解决方案和工程文件

IDE 的工作贯穿于整个工程文件和解决方案系统之中。一个解决方案由一个或者一系列为了解决某个问题的工程文件所组成。每个工程文件都有专门的元素,对要解决的问题的全部或部分进行访问。下面的示例可以更加明确地说明这个问题。比如要创建一个应用程序,来追踪在电视购物网上人们打电话购物的顺序。首先,将应用程序分解为若干比较简单的组件,比如,分解为一个处理日期的控件和一个处理货币流通的控件,并创建使用这两个控件的窗体。

在 Visual J++ 中,这个应用程序就包括 3 个独立的工程文件:

- 用来创建处理日期控件的工程文件
- 用来创建处理货币控件的工程文件
- 用来创建使用以上两个控件的窗体的工程文件

以这样的方式分别创建 3 个工程文件,就可以独立并更加灵活地应用它们。例如,如果要创建一个日记应用程序,这时,可以从头做起,也可以直接利用前面所创建的处理日期的控件。另外,这种相对独立性,也可以使你及时对控件做出必要的改动。当然,最简单的情形就是,一个解决方案中只包含一个工程文件。在本书中,大部分的示例都是每个解决方案中只包含一个工程文件的情形。

1.2.2 Java 和 Windows: WFC 和 J/Direct

目前,许多计算机都在使用 Windows 操作系统。使用 Visual J++,可以方便地利用和发挥 Windows 系统的优点,这是因为 Visual J++ 有了支持 Java 的 WFC。

WFC

在前面我们提到,Java 依赖于软件包来处理大多数函数。为了清楚地比较 WFC 和标准 Java 软件包的区别,我们专门举出下面的示例:窗口。

标准 Java 窗口软件包是 java.awt。java.awt 软件包支持的简单的控件,它由一些绘图代码组成,并支持可移植性。

WFC 窗口库在 com.ms.wfc.ui 软件包中,其中的绘图代码进行了优化设计,更适合于 Win32 操作系统,如 Windows 95、Windows 98、Windows NT 的用户编程界面。它支持包括树视图、列表视图和多文本编辑等所有的 Win32 控件。表 1-2 列出了 java.awt 和 WFC 中的控件。

WFC 还提供了操作动态 HTML(DHTML)的方法,DHTML 允许 Internet 开发者创建 Web 页和服务端应用程序,在用户从网上下载后,便对用户的 HTML 作相应的改动。

表 1-2 java.awt 和 WFC 控件

java.awt 控件	WFC 控件	java.awt 控件	WFC 控件
Button	Animation	TextField	ListView
Checkbox	Button		PictureBox
Choice	CheckBox		RichEdit
Label	ComboBox		StatusBar
List	Edit		TabStrip
Menu	ImageList		ToolBar
Scrollbar	Label		TrackBar
TextArea	ListBox		TreeView

J/Direct

J/Direct 是 Java 为本机方法设计的应用程序编程接口。当 WFC 不能提供足够的支持时,可以利用 J/Direct API 访问包括 Win32 应用程序编程接口的 Microsoft Windows 动态链接库(DLL)。

本机方法属于非 Java 方法。它之所以被称为本机方法,是因为它访问的是本地计算机,而不是 Java 虚拟机。当 Java 虚拟机无法提供对给定的某个操作的支持时,就需要使用本机方法。例如,可以使用 J/Direct 调用 Win32 API,以访问 Windows 注册表。

另外,使用 Visual J++ 中的 J/Direct Call Builder,就可以充分发挥包括 J/Direct 本身在内的优点,并快速建立包括 J/Direct 的应用程序。J/Direct Call Builder 大大简化了 Windows 环境下 Java 程序的编写。

1.2.3 Java 的可移植性和 Visual J++

Visual J++ 支持创建可移植的 Java 程序。因此,任何可移植的 Java 程序,无论是 Java 小程序、Java Beans,还是 Java 应用程序,都可以在 Visual J++ 上编译和运行。

根据本章前面对 Java 语言可移植性的介绍,我们知道,Java 的移植性是以不依赖于机器的字节码为基础的。只要提供“合适”的虚拟机,Visual J++ 所产生的字节码就可以在任何类型的计算机上运行。这里的“合适”是指支持同样的语言版本。有些虚拟机支持 1.0 版本,而另外一些虚拟机支持 1.1 版本。值得时刻注意的是,基于 1.1 版本的代码,不可能在 1.0 版本的虚拟机上运行。另外要注意,使用了 WFC 的 Java 应用程序是不可移植的。这是因为 WFC 是专门为 Win32 API 而设计的。使用了 WFC 的 Java 应用程序,只可以在 Windows95、Windows 98 和 Windows NT 等以 Win32 为基础的操作系统上运行。

1.2.4 Visual J++ 开发环境界面

下面的内容概述了 Visual J++ 的开发环境界面。启动 Visual J++ 后,屏幕上会出现

Visual J++ 的图像,接下来就是 Visual J++ IDE,然后就是典型的 New Project 对话框。你可以用这个对话框来创建新的 Visual J++ 工程文件,也可以打开已有的工程文件或者解决方案。此时,单击 Cancel 按钮,跳过该对话框,屏幕上将出现 Visual J++ 的 IDE,如图 1-1 所示。

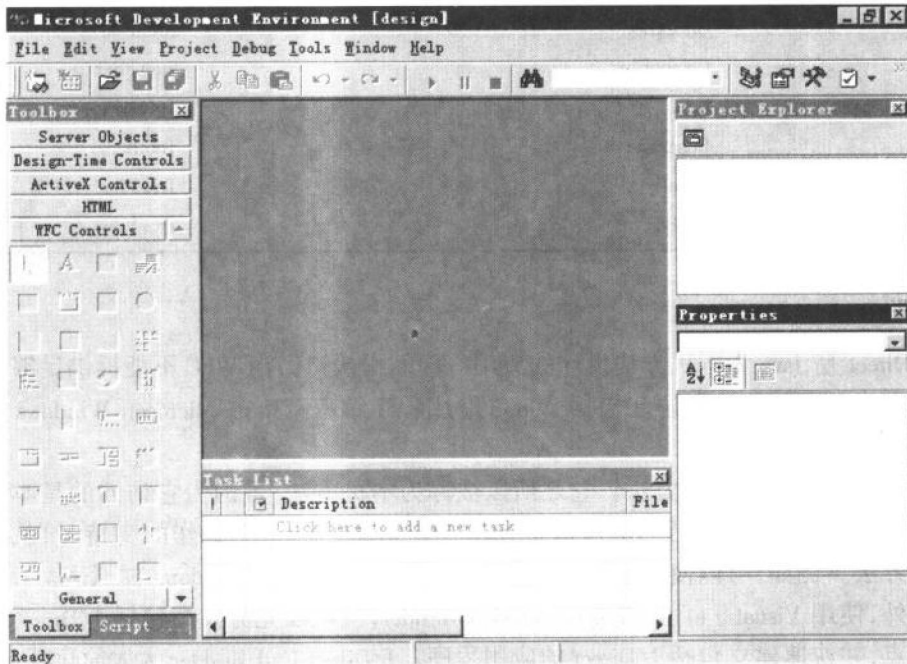


图 1-1 Visual J++ 的 IDE

图 1-2 显示了工具栏中的图标。

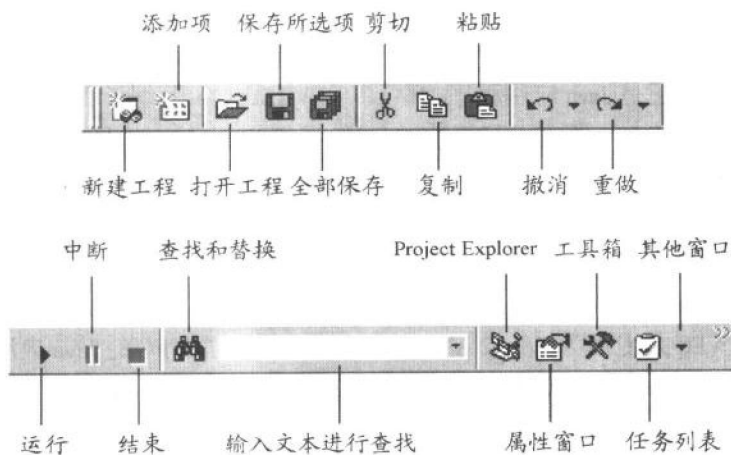


图 1-2 工具栏图标