Visual Studio 6.0 开发宝典系列丛书

Visual J++ 6.0 开发宝典

清源计算机工作室 编著

机械工业出版社

Java 语言是和 Internet 同步发展起来的一种新型网络语言，是近 20 年来计算机软件环境中的最有意义的进步之一。随着 Internet 在全世界范围内的广泛流行，以及在各个领域的渗透，Java 语言已被各行各业的人士所接受。Java 语言比 C++语言更优秀，同时又比 C++简单。Java 语言所具有的简单性、可移植性、纯面向对象、安全性和稳定性等特点，使得它成功地在全世界范围内流行。

本书共分十三章，详尽系统地介绍了 Java 语言、Visual J++ 6.0 应用程序开发环境、利用 Visual J++ 6.0 进行 Java 应用程序的开发等内容，在讲解的过程中，编排有丰富的实例程序（全书共 29 个典型实例），以帮助读者能迅速地掌握利用 Visual J++ 6.0 开发环境开发 Java 应用程序的方法和技巧。

本书既可以作为程序开发人员学习 Visual J++ 6.0 的参考书，也可以供其它工程技术人员，特别是网络软件开发者自学使用。

图书在版编目(CIP)数据

Visual J++6.0 开发宝典 / 清源计算机工作室编著.

– 北京：机械工业出版社，1999.4

(Visual Studio 6.0 开发宝典系列丛书)

ISBN 7-111-06719-3

I. V… II. 清… III. Java 语言–程序设计 IV.TP312

中国版本图书馆 CIP 数据核字（1999）第 06024 号

出版人：马九荣（北京市百万庄大街 22 号 邮政编码 100037）

责任编辑：边萌 郑文斌 版式设计：江思敏

封面设计：姚毅 责任印制：何全君

三河市宏达印刷厂印刷 • 新华书店北京发行所发行

1999 年 4 月第 1 版第 1 次印刷

787mm×1092mm1/16 • 24.5 印张 • 591 千字

0001– 5000 册

定价：41.00 元

前　言

至少掌握一门编程语言是各行各业技术人员的最基本的愿望。面对市场上层出不穷的编程软件，Microsoft Visual Studio 系列软件逐渐获得了大家的喜爱，并迅速占领了市场。作为一个软件开发人员，熟悉并掌握相关的 Visual Studio 程序开发工具是必要的。本套丛书“Visual Studio 6.0 开发宝典系列丛书”精选了目前已经得到广大用户喜爱并认可的最热门编程软件（涉及到 Microsoft Visual Studio 98 系列的所有软件），是 Visual Studio 家族中最新的产品。由于微软已经成为软件界的领导者，因此微软的系列应用程序开发软件也在应用程序领域几乎占据了统治地位，成为开发 Windows 应用程序的用户们的首选产品。

本套丛书主要有以下几个特点：

软件实用　一方面软件本身实用性强，目前在市场最流行，用户最多。本套丛书的每种软件都是针对某一相关的应用领域，软件功能强大，简单易学，用户可用它设计和开发功能强大的应用程序。软件实用的另一方面，体现在每个软件都是结合作者的实际开发经验来进行的，能带给用户最实用的开发技巧、经验等。在这套丛书里，没有高深的理论描述，只有深入浅出的讲解和简短使用的例子，可帮助读者循序渐进地掌握软件最精髓的内容。本套丛书主要面向具有一定使用基础的用户，作者精选了一些经过自己实际检验的例子，便于读者重点掌握，消化吸收，学会开发大型应用程序的方法。同时为了照顾初学者，书中还选了些短小生动的例子，可以帮助初学者尽快入门。

讲解全面　本套丛书集中全面地介绍了 Visual Studio 98 各软件的功能和开发方法，并结合大量的开发实例进行讲解。丛书的覆盖领域包括一般工程软件的开发方法、数据库、多媒体以及网络等各方面，覆盖了 Windows 编程的大部分领域。

编写质量高　本套丛书全部由清华大学的硕士和博士所写，他们都具有一定的实际开发经验，因此每本书叙述清楚、语言严谨、结构安排合理，使读者在规范化的文字叙述当中快速掌握软件的精华。

本套丛书是由相关软件使用经验丰富的作者编写，他们在编写过程中，结合自己实践经验进行讲解，内容翔实全面，具有许多参考书上没有介绍的较高层次的内容。对于初学者或有关程序设计人员具有较高的参考价值。

清源计算机工作室

1999 年

编 者 的 话

Java 是一种由 Sun 公司开发的新一代编程语言。由于 Java 运行环境与平台的无关性，所以移植很方便。不论使用的是何种 WWW 浏览器、何种操作系统和何种计算机，只要该浏览器支持 Java，就可以在上面看到各种由 Java 写成的生动的主页。Java 正在逐步成为 Internet 应用的主流开发语言，被人们美称为“网络上的世界语”。IT 界的巨头、微软的缔造者比尔·盖茨曾说过：“Java 是长期以来最卓越的程序设计语言”。它彻底改变了应用软件的开发模式，带来了自 PC 机以来的又一次技术革命。

Visual J++ 6.0 是 Microsoft 公司推出的 Java 编程开发集成环境。为满足广大编程人员对学习 Visual J++ 6.0 的要求，我们编写了本书。为利于读者更方便地学习，我们提供了一些优秀的实例。同时为适应读者循序渐进的要求，我们在内容的编排上也是由浅入深地进行介绍的。

全书共有 13 章。第 1 章介绍了 Java 的语言基础。第 2 章对 Visual J++ 6.0 进行了总体介绍，包括轻松获得 Visual J++ 6.0 的帮助和 Visual J++ 6.0 的一些特点等。第 3 章介绍了 Visual J++ 6.0 的界面和菜单系统。第 4 章介绍了利用 Visual J++ 6.0 环境创建 applet 程序和 application 程序的步骤，并进行了编译和运行。第 5 章对利用 WFC 控件实现对话框的编程方法进行了详细的讨论，并且给出了一个 Application 实例——简单计算器，同时介绍了基本 GUI 的一些方法。在第 6 章中，我们对菜单的编程方法进行了详细的讨论，并且给出了一个关于菜单的 Applet 实例——简单图形功能，同时学习了高级 GUI 和有关 Java 进行图形处理的一些方法。第 7 章介绍了 Java 语言的事件处理机制，它主要是要完成一个 Listener 接口，来实现对相应事件的监听和处理。第 8 章中，我们首先对 Visual J++ 6.0 的程序调试器进行了介绍，接着讲解了 Java 异常的基本概念，分析了 Java 语言的异常处理机制特点。第 9 章通过丰富的实例，介绍了利用 Java 语言进行多媒体设计的技术，如图像的装载、缩放等图像处理技术、动画技术和如何播放声音等。第 10 章详细地介绍了 Java 语言的多线程编程问题。第 11 章介绍了如何利用 Java 编制网络应用程序，Java 语言提供的网络编程机制主要有两种类型：通过 URL 和通过 socket 读取网络资源。第 12 章对如何用 Java 语言提供的各种类来编制数据库应用程序进行了详细的介绍。在第 13 章中，我们首先介绍了有关 JavaScript 的内容，然后主要讨论了 JavaScript 与 Java 之间的混合编程技术，使用 Java、JavaScript 和 HTML 的综合编程，可以写出一些漂亮的 WEB 程序。

本书由杨智明与张晓海执笔。由于水平和时间有限，错误和不妥在所难免，敬请广大读者批评和指正。

编者

1999 年

目　录

第 1 章　Java 语言编程基础

Java 语言是和 Internet 同步发展起来的一种新型网络语言，是近 20 年来计算机软件环境中的最有意义的进步之一。它是在 1995 年春季由 SUN 公司发布，随后立即得到了各 WWW 厂商的大力支持，纷纷在浏览器上加入用 Java 编写的小应用程序 Applet，迅速通过 Internet 在世界各地进行传播，它在网络中的地位同超文本链接标注语言 HTML（Hypertext Markup Language，是一种在 World Wide Web 上实现静态文本的语言）一样重要。随着 Internet 在全世界范围内的广泛流行，以及在各个领域的渗透，Java 语言已被各行各业的人士所接受。通过 Internet 的环境，利用 Java 开发应用程序可以实现各种分布式计算，进行广域范围内的合作。

Java 语言是一种强有力的编程语言，它将面向对象的设计用一种简单和熟悉的语法根植在增强的、易用的环境中。Java 有一组丰富的对象类，使程序员可以对许多公用的系统功能如窗口操作、网络和输入/输出进行简明的抽象处理。Java 具有让任何人使用 Applet 程序的能力，Applet 是一个小巧、安全、动态、跨平台、活跃、网络化的应用程序。

本章的主要内容为：

- Java 是卓越的编程语言。
- 最简单的 Java 应用例子。
- Java 词法。
- Java 基本数据类型。
- Java 运算符和表达式。
- Java 流程控制。
- Java 数组。
- Java 类。
- Java 接口和包。
- Java API 简介。

1.1　Java 是卓越的编程语言

IT 界的巨头、微软的缔造者比尔·盖茨曾说过："Java 是长期以来最卓越的程序设计语言"。那么 Java 的一些特点在哪呢？本节将做详细介绍。

1.1.1　Java 技术特点

Java 能迅速在全世界传播，是因为它拥有一些优秀的技术特点，主要有以下几个方面。

1．简单性

Java 语言最先的设想是为家用电器产品设计的，便于推广，所以被设计得简单而高效，生成的可执行代码也非常短小精悍。另外，Java 从 C++而来，使得原来是 C++的程序员很容易转向 Java；Java 取消了 C++中的一些比较复杂的部分，所以它比 C++简单，更容易学

习，其程序的可读性增强。例如，C++中的结构、联合和类的概念重合之处很多，Java 语言只保留了类的概念。关于 Java 和 C++一些不同点的比较将在后面介绍。

2．可移植性

Java 定义出自己的一套虚拟机，以及这套虚拟机上所使用的机器码——JavaBytecode。Java 通过预先将源代码编译为接近于机器指令的字节码，有效地克服了传统解释型语言的性能瓶颈；又由于解释执行 JavaBytecode 只需要 Java 运行系统不需要具体某个平台的支持，因此，这一点保证了 Java 运行环境与平台的无关性，所以移植很方便。Java 语言的可移植性主要是由以下两条机制保证的：

●Java 从本质上是解释执行的，任何一种平台，任何一台机器，只要配备了 Java 解释器—Java 虚拟机，就可以运行 Java 程序。

●Java 的数据类型在任何机器上都是一致的，它不支持依赖于特定硬件环境的数据类型。

Java 语言的可移植性具有深远的意义，它开创了程序设计的新时代。它不仅使软件开发者实现了"一次性开发，永远运行"的方便，而且迎合了网络分布式计算的思想。在 Java 出现以前，Internet 只是被人们当作一个巨大的硬盘，里面有无数的静态信息，人们把各种信息发布到 Internet 上，同时可以从 Internet 上得到自己希望得到的信息；Java 语言出现后，Internet 则成为了一个巨大的操作系统，而 Java 就是这个系统的语言。Java 语言的出现，使得分布式计算成为了可能。例如：用 Java 语言编写的任何程序如字处理程序，存放在某台服务器上，该服务器与 Internet 相连，这样，与 Internet 相连的任何用户只要得到许可，就可以下载并运行这个程序，这样大大节省了开发时间和存放程序的硬盘空间，同时也省去了管理的麻烦；又如用 Java 编写的一个模型仿真程序，它发布在与 Internet 相连的服务器上，用户在客户端下载该模型仿真程序，并根据自己的条件参数进行仿真模拟，同时可以把结果返回给服务器。

Java 语言给未来的软件带来了新的思路，它改变着人们的生活方式、科学研究合作的环境。Java 语言的可移植性使得它成为未来网络世界的"通用语言"。

3．纯面向对象

Java 是一种完全面向对象的程序设计语言。Java 程序代码以类的形式组成。Java 抛弃了 C++中非面向对象的特性。面向对象的程序设计是 Java 语言的一个极为重要的特点，在后面将进行更为详细的介绍。

4．结构中立性

Java 采用了完全统一的语言文本，如 Java 的基本数据类型不会随机器的变化而变化，如一个整型总是 32 位，而不像 C++那样，随着机器的不同，整型变量的长度是不相同的。同时 Java 语言环境还定义了一个用于访问底层 OS 功能的扩展类库，以使 Java 的应用程序能不依赖于具体系统。Java 语言的结构中立性是保证 Java 的可移植性的基础。

5．分布性

Java 支持 WWW 客户机/服务器计算模式。通过 Java 提供的类库可以处理 TCP/IP 协议，用户可以通过 URL 地址在网络上访问对象。所以，Java 是一门适合 Internet 和分布式环境

的技术。

6．稳定性和安全性

稳定性和安全性是相关的。分布式计算环境要求软件具有高度的稳定性和安全性。Java 不支持指针数据类型，也不允许直接对内存进行操作。Java 还提供了内存管理机制，即一个自动的“垃圾回收”功能。Java 语言除了在语言本身上加强了安全性外，其运行环境还提供了四级安全性保障机制：

- 字节码校验器。
- 类装载器。
- 文件访问控制。
- 运行时内存布局。

可以看出，Java 语言比以往的任何程序设计语言都注重安全性和稳定性，这也是它成为网络语言、适应分布式网络计算所需要的。

1.1.2　Java 和 C++的不同点比较

C++适应了软件工程界的面向对象的新潮流，但是 C++并不能真正满足面向对象的编程，它保留了很多非面向对象的特点。后来创建的 Java 语言实现纯面向对象的特点，使得它更为简单，更为人们所接受。所以现在很多的 C++程序员纷纷转向 Java 程序设计，一方面是全球网络大环境下工作和学习的需要，另一方面也正是因为它与 C++相比显得简单易学。

Java 比 C++更优秀，因为它排除了它不需要的部分。从 Java 语言的起源与发展来看，一方面，它由 C++发展而来，其语言风格与 C++十分相似；另一方面，Java 又比 C++简单，它抛弃了 C++中一些不是绝对需要的内容。这些内容主要有以下几部分：

1．指针

指针或内存地址是 C++中最有效也是最有害的特性。不正确的指针操作会引起许多错误。在 Java 中不允许直接使用指针。

2．内存分配

C++中的内存分配与指针操作有同样的危险。其内存分配是通过 malloc()和 free()库函数以及 new 和 delete 两个运算符来实现的。程序员需要自己释放空间。Java 中没有 malloc 和 free 函数。由于每个复杂的数据结构都是对象，它们通过 new 运算符在内存堆上分配空间。一旦不再访问对象，占据的内存空间就会被回收。因此根本不需要 free 和 delete。这种技术被称为“垃圾回收”。

3．全局变量

C++中全局变量作为程序的状态信号没有很好地进行封装。Java 中，只有类是全局的。不可能创建一个不属于任何类的全局变量。

4．脆弱的数据类型

不同的 C++编译器根据不同机器的实际配置分配给数据类型以不同的字长，如 16 位、32 位或 64 位。而在 Java 中，所给定的基本数据类型确定一个合理的字长并保持不变。Java

解释器的这种严格与硬件无关的数据类型很难对代码进行优化和实现。但这是唯一能够保证跨平台实现的途径。

5. Goto

在 C++引入异常处理以前，goto 经常被用来在异常处理中跳出循环。Java 没有 goto 语句，Java 中严格定义的异常处理机制使 goto 没有再存在的必要，取消这种随意跳转的语句有利于优化代码以及保持系统的稳定性和安全性。

6. 分离的头文件

C++有头文件，而 Java 中没有头文件。

7. Java 不直接支持多重继承

C++中具有多重继承的特性。

1.1.3 面向对象编程的特征

在现实生活中，人们通过抽象来管理复杂事物。这种将复杂事物进行分层抽象的方法不仅适用于现实世界，而且也适用于计算机程序。长期以来，人们在解决问题的同时一直在寻求与问题本身结构上尽可能一致的方法，使得人们在分析、设计和实现程序过程中，与人们认识事物的过程尽量一致，因此产生了面向对象的程序设计方法。当前，面向对象是计算机领域最流行的程序设计方法。在面向对象方法中，传统的算法程序可以抽象成各种要处理的对象，一系列处理步骤可构成独立的对象间的信息集。在每个对象中封装了自己的处理问题的方法和行为。我们将这些对象甚至抽象对象当作现实世界具体的实例，它们能响应外界的刺激并进行相应的动作。这就是面向对象编程的基础。正像人类理解复杂事物的方式一样，面向对象的概念构成了 Java 的核心。面向对象具有封装、继承、多态三个主要特性。

1. 封装

从最基本的角度看,任何程序都包含两部分：代码和数据。在传统的代码模型中，数据在内存中被分配并由子程序或函数中的代码来处理。而面向对象设计的核心一环是将处理数据的代码和数据的声明和存储封装在一起。

可以把封装认为是一个将代码和数据保护起来的保护膜。这个保护膜定义了对象的行为并且保护代码和数据不被任何其它代码任意访问。即一个对象中的数据和代码相对于程序的其余部分是不可见的，它能防止那些非期望的交互和非法的访问。

在 Java 中，封装的基本单元是类。你可以创建一个类，它是一组具有行为和结构的对象的一种抽象。对象是具有行为和结构的类的具体实例，就好象将类当作模子造出的一个翻版。因此，有时称对象为类的实例。

封装的目的是为了减少复杂性。因此，在类中具有隐藏复杂性的机制。类中的每个方法或变量都可以被定义为公有或私有。类的公有部分可以让外界的用户知道或必须知道，公有的实例变量和方法是一个类与外部的接口，程序的其它部分通过这个接口使用类的功能。而定义成私有的方法和实例数据则不能被类外的其它代码所访问。

改变类的其它部分不会对整个程序产生非预期的影响。只要保持类接口不变，该类的内部工作方式可以随便改动。

Java 不仅允许程序设计者自己创建类，还提供了大量适合于各种应用的预定义类库。例如：有关屏幕显示、文件访问、数学计算等方面的类库。

2．继承

通常我们中的大部分人都会将世界看成由相互关联的具有层次的对象组成，如动物、哺乳动物和狗。哺乳动物看作是动物的继承，而狗又是哺乳动物的继承。下一层次是上一层次的进一步具体描述，并且下一层次继承了上一层次的所有特性。一个多层次的继承关系构成了一个类树结构。

在面向对象的程序设计中，继承指在已有类的基础上建立一个新类。新类自动拥有父类的所有元素：方法和实例变量，然后再根据需要添加新任务所需的方法和（或）实例变量。

合理使用继承可以减少很多的重复劳动。如果在一个类中实现了一个特别的功能，那么在它的派生类中就可以重复使用这些功能，而不需要重新编程来实现。对 Java 的内置类可以创建派生类，也可以对自己创建的类建立派生类。

一个不由其它类派生来的类称为基类；一个派生类的最近父亲（其最近的上层类）叫做该类的父类；从某一类派生出来的类叫做该类的子类。并不是子类的层次越多越好，如果层次太多的话，不如重新创建一个新类。

一个给定的类从派生它的基类到它自身可以经过好多个层次。一个类不仅能继承它的父类的所有的方法和实例变量，而且还能继承从它的基类开始到它自身之间的所有经过的层次上的类的方法和实例变量。

Java 不支持多重继承，即指一个类不能有一个以上的父类。

继承和封装具有很好的合作性。如果一个给定的类封装了某些属性，那么任何子类将继承这些属性并增加了自己特有的属性。

3．多态

对象中的方法通过参数传递信息。这些参数作为函数的输入值需要方法加以执行。

为了在大部分函数性语言编程中完成两个不同的任务，需要给两个函数定义不同的名字。多态意味着一个对象具有多个面孔，允许一个方法根据传递过来的参数类型采用不同的实现。比如，可能有两个方法都交 print()，一个方法是在屏幕上显示字符串，另一个方法则是在屏幕上显示一个位图。当调用 print()时，到底激活哪一个函数取决于调用的参数是字符对象还是位图对象。

多态性有时也指方法的重载。方法的重载是指同一个方法名在上下文中有不同的含义，是让类以统一的方式处理不同数据类型的一种手段。它是静态的，这是因为在实现类并在编写方法之前要考虑到将要遇到的所有数据类型。在某种程度上，这是非常必要的而且也能导致清晰和可预料的代码，然而它不灵活，经常需要在许多代码被冻结和没有源代码的情况下扩充环境。子类提供了更丰富的动态运行时的多态性。

在本节中，我们指出了 Java 的一些优秀的技术特性，这些 Java 的技术特性使得它能在全世界迅速传播；对 Java 和 C++两种编程语言进行了比较；同时讨论了面向对象的程序设计的一些概念，面向对象的三个主要特性：封装、继承和多态。

1.2 最简单的 Java 应用例子

Java 源文件是由类定义组成的，Java 源文件中只能以类定义的形式来进行编程。一个 Java 源文件中可以包括一个类定义也可以包括多个类定义。Java 源文件文件以.Java 为后缀名保存。当 Java 源代码被编译后，每个类分别保存在以类定义的带有.class 后缀名的输出文件中。这样，一个 Java 源文件经过编译后可以生成多个类文件。由于 Java 没有全局函数或变量，Java 源文件只保存有一个或多个类定义，这种限制导致小程序更复杂。

在本节中，我们立即开始编写、编译并运行一个标准的"Hello Java!"应用程序和一个可嵌入 HTML 文件的，并由 WWW 服务器解释执行的 Applet 程序（小应用程序）。

1.2.1 应用程序实例

应用程序是可以用 Java 解释器直接执行的程序。

典型实例 1 简单应用程序

```
// 文件名：HelloJava.java
class HelloJava
{
    public static void main(String[] args)
    {
        // 打印一字符串
        System.out.println("Hello Java!");
    }
}
```

把文件命名为 HelloJava.java。要注意把文件名与类名取成一样，且两者的大小写要一致。

接着编译这个文件，就需要运行 Java 的编译器 javac，并在命令行输入源文件名，如下所示：

```
C:\>javac HelloJava.java
```

Javac 编译器会产生名为 HelloJava.Class 的与处理器无关的二进制字节码文件—bytecode 程序。用下列方法运行此文件：

```
C:\>java HelloJava
```

在屏幕上会有如下的输出结果：

```
Hello Java!
```

这样一个最简单的 Java 应用便写成了，就这么简单。下面对这个程序进行分析。

●首先要使用关键字 class 来定义一个新的类，它为一个公共类（public class），此处类名为 HelloJava，整个类体包含在{}中。

●用"//"号来进行注释，注释部分仅仅为程序起一个说明的作用，它们在词法分析前被

去掉。

●该类定义了一个 main（）方法，其中关键字 public，表示所有的类都可以使用该方法。对于一个应用程序来讲，必须要有一个 main（）方法，该方法用来表示程序执行的入口点。

●该 main（）方法前有关键字 static，表明该方法是一个静态方法。

●void 关键字表示 main（）方法没有返回值。

●main（）方法中只有一条语句：

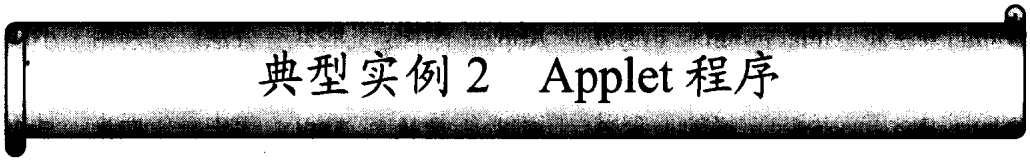
System.out.println("Hello Java!");

用来输出一个字符串，它的功能与 C 语言中的 print（）函数相同。

●Java 程序的每条语句必须要用"；"结束。

●一个 Java 程序可以有多个类，每个类可以有多个方法，但是最多只有一个公共类和一个 main（）方法。

1.2.2　小应用程序 Applet 实例



小应用程序可以嵌入在 HTML 中，并由 WWW 浏览器来解释执行的程序，它不能象应用程序一样可以用 Java 解释器直接执行。

```
// 文件名：HelloWWW.java
import java.applet.Applet;
import java.awt.Graphics;
public class HelloWWW extends Applet
{
    public void init()
    {
        resize(400,300);
    }
    public void paint(Graphics g)
    {
        g.drawString("Hello WWW!",60,30);
    }
}

// 文件名：HelloWWW.html
<HTML>
<HEAD>
```