Intel Visual Fortran
GAOJI CHENGXU SHEJI

Intel Visual Fortran
高级程序设计

李兴田　张丽萍／编著

中国铁道出版社有限公司
CHINA RAILWAY PUBLISHING HOUSE CO., LTD.

Intel Visual Fortran
高级程序设计

李兴田　张丽萍　编著

中国铁道出版社有限公司
CHINA RAILWAY PUBLISHING HOUSE CO., LTD.

内 容 简 介

本书系统介绍了用 Intel Visual Fortran 进行视窗创建、GDI 绘图、OpenGL 图形、多线程编程以及混合编程等多个主题的程序设计。全书共分 9 章，首先对 Fortran 语言及其编译器做了简单的介绍；然后系统介绍了 QuickWin 窗口程序设计；接着在阐述 Windowing 程序设计的基础上，进一步介绍了 OpenGL 图形程序设计和 Intel Visual Fortran 多线程编程；最后简单介绍了混合语言编程。

本书语言简洁，实例丰富，适合用 Fortran 语言进行程序设计和开发的大学生及其他读者学习，也可作为从事 Fortran 教学研究、开发及应用方面的工程技术人员的参考书。

图书在版编目（CIP）数据

Intel Visual Fortran 高级程序设计/李兴田，张丽萍编著. —北京:中国铁道出版社有限公司，2019.4
ISBN 978-7-113-25666-1

Ⅰ.①I… Ⅱ.①李… ②张… Ⅲ.①FORTRAN 语言-程序设计 Ⅳ.①TP312.8

中国版本图书馆 CIP 数据核字(2019)第 059122 号

书　　名：Intel Visual Fortran 高级程序设计
作　　者：李兴田　张丽萍

策　　划：曾露平
责任编辑：曾露平　彭立辉
封面制作：刘　颖
责任校对：张玉华
责任印制：郭向伟

出版发行：中国铁道出版社有限公司（100054，北京市西城区右安门西街 8 号）
网　　址：http://www.tdpress.com/51eds/
印　　刷：北京虎彩文化传播有限公司
版　　次：2019 年 4 月第 1 版　2019 年 4 月第 1 次印刷
开　　本：787 mm×1 092 mm　1/16　**印张**：17　**字数**：414 千
书　　号：ISBN 978-7-113-25666-1
定　　价：45.00 元

PREFACE

前 言

Fortran 是世界上最早出现的一种计算机高级程序设计语言，广泛应用于科学和工程计算领域。由于其自身的优势和人们对计算速度无止境的追求，Fortran 语言在过去经历了巨大的成功和辉煌后，今天在科学计算领域中仍占有非常重要的地位。

由于简洁的语法、高效的运行速度等优势，不少工科专业都选择开设 Fortran 语言编程课程。但是，限于课时，多数 Fortran 语言教学只选择 Fortran 标准程序设计而舍弃语言扩展，这在以后的学习和工作中是远远不够的。

如果编制实现了完善的数值计算程序，那么图形用户界面是必要的，因为这将使程序的使用更加友好；如果程序的输入/输出均需要数据存储和可视化，则数据库和绘图功能是非常重要的；当工程计算量较大，而计算机有充足的硬件设备可供利用时，那么开辟线程进行计算加速是非常值得的。如果为此而选择其他编程语言，则不但需要投入大量的时间重新学习一门编程语言，还要面对混合编程中编码不便、调试困难等情况，其不足是显而易见的。

正因如此，本书不再讨论 Fortran 语言程序设计基础，而是对多个相关主题进行讨论，使读者专注于科学和工程数值计算的同时，尽可能地用一种语言完成开发工作，这对效率也是一个促进和提升。

本书的主要内容包括 Fortran 的发展历史、Fortran 编译器、QuickWin Application 基础、QuickWin 用户交互、QuickWin 基本控件的使用、图形程序设计、Windowing 程序设计、OpenGL 图形程序设计、Intel Visual Fortran 多线程编程以及混合语言编程。

由于时间仓促，编者水平有限，我们虽然做了很大的努力，但书中仍难免存在疏漏与不妥之处，请读者不吝赐教。对您的帮助与支持，我们深表感谢。

本书的出版得到了兰州交通大学“百人计划”、兰州交通大学土木工程学院“BIM 教学团队”项目的资助，在此表示诚挚的感谢。

编者

2019 年 1 月

CONTENTS

目 录

绪论

Fortran 的发展历史

FORTRAN 语言是世界上第一个被正式推广使用的计算机高级语言，FORTRAN 是 Formula Translation 的缩写。顾名思义，这种语言适合科学计算，用它编写的程序接近数学中的表达式，具有良好的可读性。FORTRAN 语言自 1954 年提出以后，至今已有 60 多年的历史，但仍经久不衰。在历经多种版本的演变后，将 FORTRAN 变为了 Fortran。时至今天，Fortran 语言始终是科学计算领域常用的计算机高级编程语言。

在展开相关的主题讨论之前，先了解一下 Fortran 语言的发展演变史。

1. FORTRAN Ⅰ

1954 年，在 IBM704 大型计算机发布之前，John Backus 和他所在的 IBM 中的小组就开始了 FORTRAN 语言的开发工作。他们宣称，FORTRAN 语言将提高手工编码程序的效率。

由于当时的计算机以科学计算为主，其内存都很小，运行速度很慢且不可靠，这就直接决定了早期 FORTRAN 版本的特点：能够生成高速度的目标代码。

在他们的努力下，1957 年发布了 FORTRAN Ⅰ编译器。当时的 FORTRAN 语言只有几条简单的控制语句，而且主要针对于 IBM704 计算机。但是仅仅用了一年时间，该计算机上的代码约有一半是用 FORTRAN 语言编写的。毫无疑问，这是个巨大的成功。

2. FORTRAN Ⅱ

1958 年，FORTRAN Ⅱ编译器发布。该系统修正了 FORTRAN Ⅰ编译器中的许多缺陷，增加了可以单独编译子程序的功能，使较大规模的程序开发有了可能。

3. FORTRAN Ⅲ

由于存在着严重的缺陷，该版本从未正式推出。

4. FORTRAN Ⅳ

由于 FORTRAN 的流行，许多计算机厂商都为自己的机器实现了定制版本，结果导致在一种平台上写成的程序不能运行在其他平台上，由此出现了严重的不兼容问题。IBM 也意识到了这个问题，他们移除了语言中依赖于硬件的特性，并在 1961 年正式推出 FORTRAN Ⅳ编译器。

5. FORTRAN 66 标准

在 20 世纪 60 年代初期，随着 FORTRAN 语言使用规模的扩大，人们越来越需要一种能够脱离于任何计算机硬件环境的 FORTRAN 语言。1962 年，美国标准协会成立 FORTRAN 标准化委员会，将 FORTRAN Ⅳ标准化为 FORTRAN 66。

这个标准在科学计算领域内具有里程碑式的意义。

6. FORTRAN 77 标准

在 FORTRAN 66 标准推出之后，各家编译器厂商不断推出更具扩充性的 FORTRAN，这使得 ANSI 于 1969 年开始着手于 FORTRAN 66 标准的修正工作，最后推出新的标准规格，也就是著名的 FORTRAN 77。

这是一个非常流行的语言版本，以至于到今天，我们还能看到很多用 FORTRAN 77 编写的程序。

7. Fortran 90

随着编程语言的发展，后来的加入者（如 C 语言）允许程序员动态分配内存和定义数据结构，但是 FORTRAN 77 却无能为力。为了解决这些问题，FORTRAN 标准委员会准备推出新的标准，从而解决相关问题。但是直到 20 世纪 80 年代末，新的标准迟迟不能发布。新版本最终在 1991 年作为标准由国际标准组织出版，并且非正式地被称为 Fortran 90。

相较于上一版本，Fortran 90 加入了动态数组、指针和模块。在语法方面，Fortran 90 去掉了代码所需的固定形式，增加了自由格式，使代码格式不再死板。同时，将语言名称的拼写由 FORTRAN 变为 Fortran。

同时，微软公司将 Fortran 90 无缝集成在 Developer Studio 集成开发环境之中，推出了 Microsoft Fortran PowerStation 4.0，使 Fortran 90 真正实现了可视化编程，彻底告别了传统 DOS 环境，转到了现代的 Windows 环境。

由于微软公司为 Fortran 提供了 Win32 API、OpenGL 等函数库的接口，使得用 Fortran 语言编写可视化界面及图形绘制成为可能。虽然这些特征由平台制约，但是在 Windows 使用环境中，无疑为 Fortran 语言提供了巨大的拓展空间。

8. Fortran 95/2003/2008/2018

Fortran 90 之后，较为重要的变化发生在 Fortran 2003 中。该版本加入了面向对象的语言特征、与 C 语言的互通性等。其他版本都是在前一版本的基础上进行小幅修订。

今天，当计算机工业发生了深刻的变化时，Fortran 仍然在发展壮大着。当各种编程语言争奇斗艳时，在数值计算领域中，仍然可以看到 Fortran 的王者之风。

第1章 Fortran 编译器

在进行窗口程序设计之前，先了解一下 Fortran 编译器的简介、Intel Visual Fortran 编译器的安装和基本使用方法。

1.1 Fortran 编译器简介

Fortran 是一种高级计算机编程语言，所有 Fortran 程序源代码必须经过编译器的编译、连接，才能被翻译成计算机所能识别的机器码，从而完成源程序设置的任务。

支持 Fortran 语言的编译器较多，PC 中 Windows 平台上常见的有以下几种：

（1）Fortran PowerStation 4.0：微软公司将 Fortran 90 编译器集成到 Developer Studio 开发环境中之后推出的 Fortran 编译器，真正实现了 Fortran 的可视化编程。

（2）Digital Visual Fortran：微软公司和数据设备公司（DEC）联合开发的功能更强大的 Fortran 编译器。

（3）Compaq Visual Fortran：数据设备公司和康柏公司（Compaq）合并后推出的 Fortran 编译器以及康柏公司并入惠普后推出的新版本编译器。

（4）Intel Visual Fortran 惠普公司将 Windows 下的编译器转售给 Intel 公司后，由 Intel 公司开发的 Fortran 编译器，目前的新版本是 Intel Parallel Studio XE 2018。

（5）Salford Fortran 由 Silverfrost 公司开发的 Fortran 编译器，可用于 Win32 平台和.net 平台上应用程序的开发，支持完整的 Fortran 95 语法和部分 Fortran 2003 语法。

除此之外，还有一些较为常见的 Fortran 编译器，如 Lahey Fortran、Absoft Fortran 和 OpenWatcom 等。

从国内目前的使用情况来看，前 4 个编译器在 Windows 平台上最为常见。考虑到 Fortran 编译器目前的发展情况，本书所有的程序都基于 Intel Parallel Studio XE 2018 编译器生成。

同 C、C++语言一样，Fortran 语言本身没有提供图形界面输出功能。为了用 Fortran 语言编写 Windows 图形界面程序，必须借助编译器提供的扩展功能。Intel Visual Fortran 编译器几乎封装了完整的 Windows API 和 OpenGL 函数，提供了良好、稳定的编程接口。所以，利用 Intel Visual Fortran 编译器，Fortran 语言不但能用于数值计算，还能进行界面和图形编程。

1.2 Intel Visual Fortran 编译器的安装和使用

为了用 Fortran 语言编写窗口程序，首先要安装 Intel Visual Fortran 编译器。总体操作步骤如下：

（1）在 Windows 10 系统上安装 Visual Studio 2015。

（2）安装 Intel Parallel Studio XE 2018 编译器，并选择集成到 Visual Studio 2015。

总之，Windows 平台上软件的安装过程比较简单，请读者自行完成。如果需要，可以利用搜索引擎获取相关的安装帮助。

成功安装 Intel Parallel Studio XE 2018 编译器后，就能编译 Fortran 程序。具体操作步骤如下：

（1）启动 Visual Studio 2015，选择 File→New→Project 命令，弹出如图 1.1 所示的对话框。

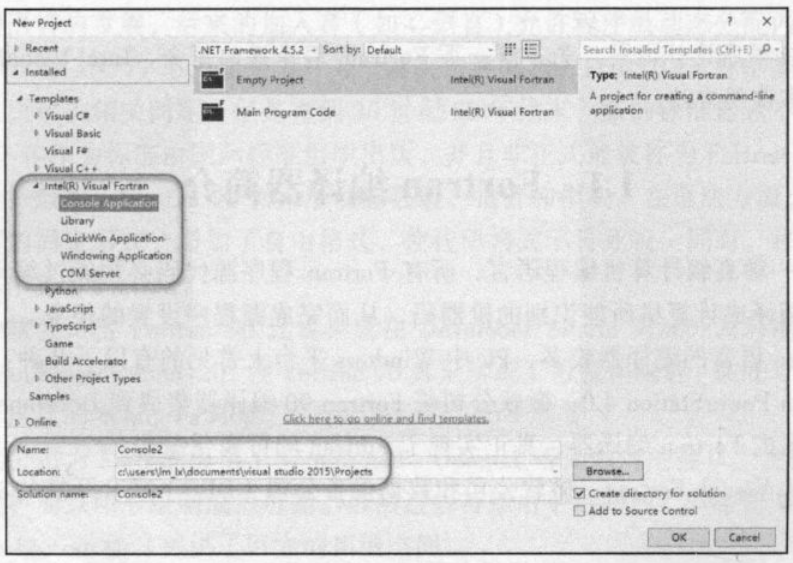

图 1.1　新建项目对话框

（2）选择 Fortran 项目类型和模板，然后指定项目名称及其保存位置，最后单击 OK 按钮完成新项目的创建。

（3）在弹出的对话框中，右击 Source Files 文件夹，如图 1.2 所示。

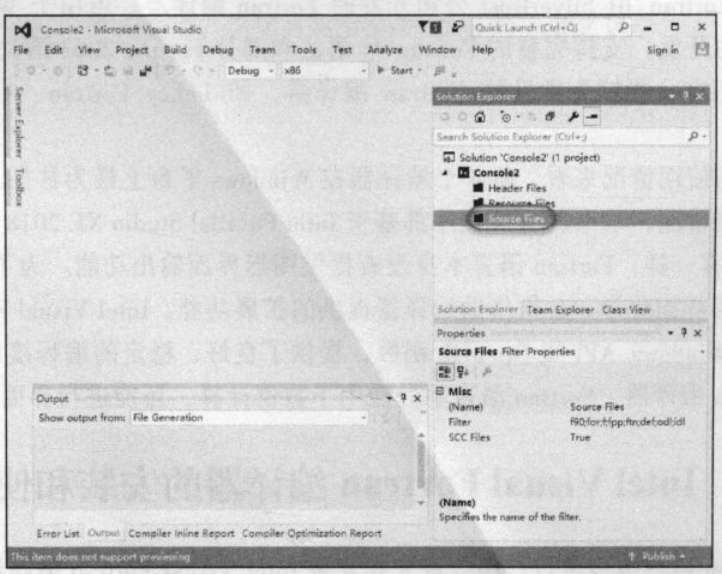

图 1.2　右击 Source Files 文件夹

（4）在弹出的如图 1.3 所示的快捷菜单中，选择 Add→New Item 命令，如图 1.4 所示。

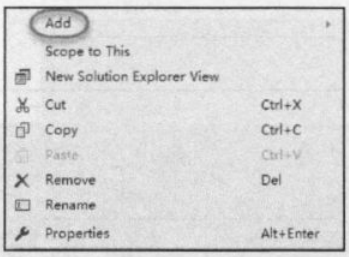

图 1.3　选择 Add 子菜单项

图 1.4　选择 New Item 命令

（5）在弹出的如图 1.5 所示的对话框中，选择 Fortran Free-form File 文件格式，指定程序源文件名称及其保存位置，最后单击 Add 按钮完成程序文件的创建。

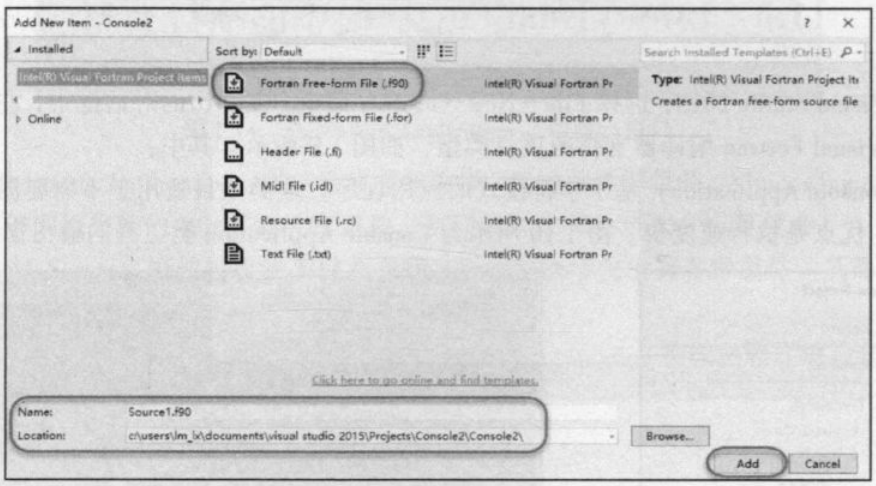

图 1.5　选择 Fortran Free-form File

（6）回到 Visual Studio 2015 主界面，在程序编辑器中输入如图 1.6 所示的简单代码。

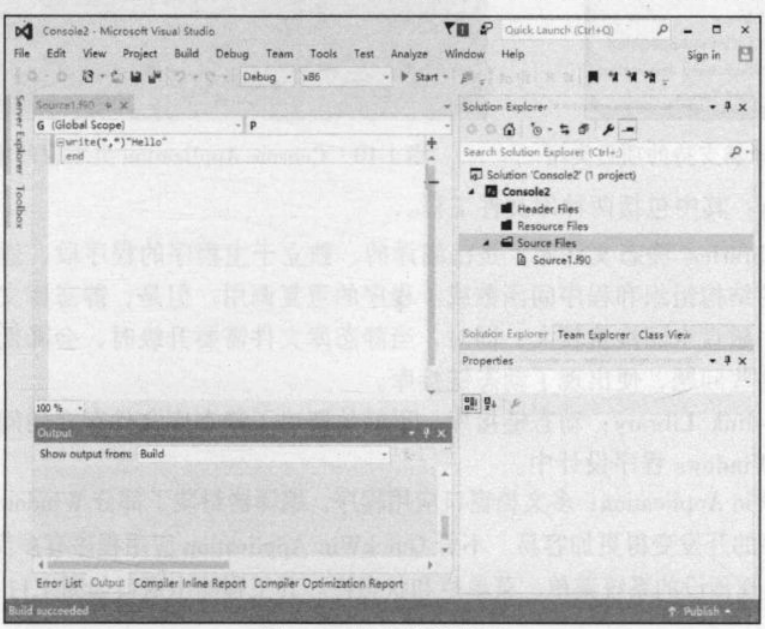

图 1.6　输入简单代码

（7）选择 Debug 菜单中的 Start Without Debugging 命令（见图 1.7），或者按下【Ctrl+F5】组合键编译并运行程序。

（8）输出结果如图 1.8 所示。

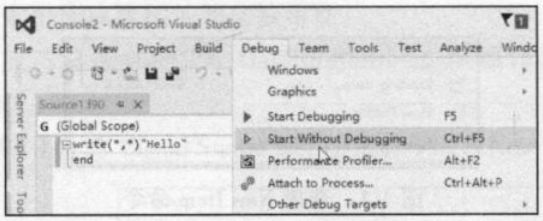

图 1.7　编译并运行程序

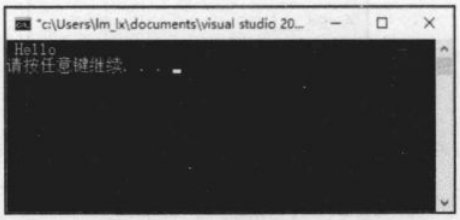

图 1.8　程序运行结果

1.3　Intel Visual Fortran 中的项目类型

启动 Visual Studio 2015，选择 File→New→Project 命令，在弹出的对话框中，左侧列表显示了 Intel Visual Fortran 编译器支持的项目类型，如图 1.9 所示。其中：

（1）Console Application：基于字符模式的应用程序，这类项目适用于不需要图形输出的程序设计，优点是执行速度快。图 1.10 所示为 Console Application 类工程的输出窗口。

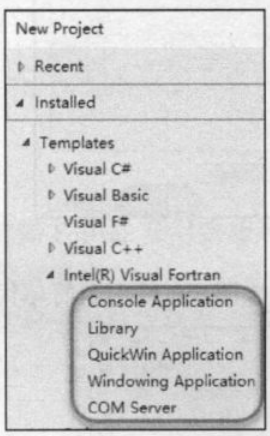

图 1.9　编译器支持的项目类型

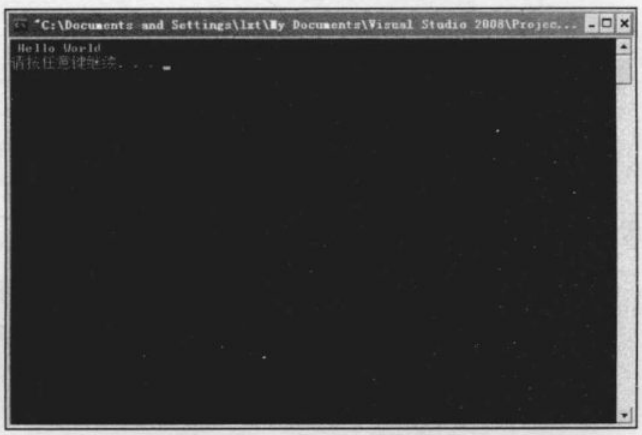

图 1.10　Console Application 类工程的输出窗口

（2）Library：其中包括两种库文件工程。

- Static Libraries 静态文件库，是已编译的、独立于主程序的程序段，适合于大型项目中程序的结构组织和程序间函数或子程序的重复调用。但是，静态库文件会插入程序调用点，磁盘空间浪费较大；同时，当静态库文件需要升级时，会降低工作效率。为了解决这些问题，便出现了动态链接库。
- Dynamic-link Library：动态链接库，很好地解决了静态库文件存在的问题，所以广泛使用在 Windows 程序设计中。

（3）QuickWin Application：多文档窗口应用程序。编译器封装了部分 Windows API 函数，使 Fortran 窗口程序的开发变得更加容易。不管 QuickWin Application 应用程序有多简单，默认情况下窗口中总会出现预设的系统菜单、菜单栏和状态栏。其工程输出窗口如图 1.11 所示。

图 1.11　QuickWin Application 工程输出窗口

（4）Windowing Application：窗口界面应用程序，经过编译器封装。Windowing Application 应用程序能够调用完整的 Windows API 函数，所以其开发弹性更大、也更复杂。

（5）COM Server：组件对象模型服务程序，可用模块导向编写各种组件，从而以组件组装的方式完成应用程序的开发。

本书主要讨论 QuickWin Application 和 Windowing Application 类应用程序的设计开发。

第2章

QuickWin Application 基础

本章将利用 QuickWin 工程来创建简单的窗口程序。相比常见的命令行程序，图形用户界面更加友好。

2.1 最简单的窗口程序

如果以前的程序都是基于 Console 工程类型，那么，从现在开始，让我们舍弃“黑屏”，一起体验窗口程序设计的乐趣。

跟其他语言一样，最简单的程序总是“hello world!”的例子。这是第一次创建 Fortran 窗口程序，所以，下面将给出较为详细的步骤，逐步引导读者创建第一个窗口程序。具体操作步骤如下：

（1）启动 Visual Studio 2015。

（2）选择 File→New→Project 命令，弹出如图 2.1 所示的对话框。

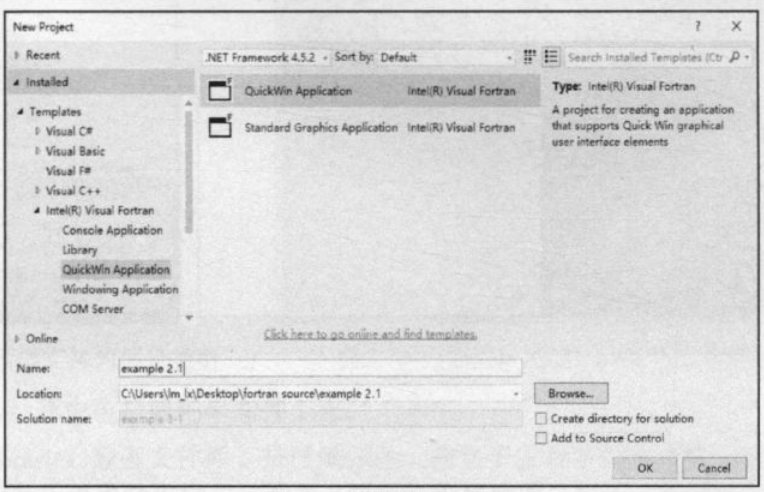

图 2.1 选择项目类型

（3）在“项目类型”和“模板”中选择 QuickWin Application，指定项目的保存位置和名称，然后单击 OK 按钮。

（4）这时，编译器界面中出现如图 2.2 所示的解决方案资源管理器。

（5）右击 Source Files 文件夹，在弹出的快捷菜单中选择 Add→New Item 命令，弹出如图 2.3 所示的对话框。

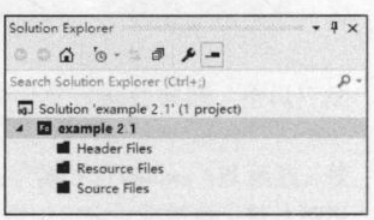

图 2.2 解决方案资源管理器

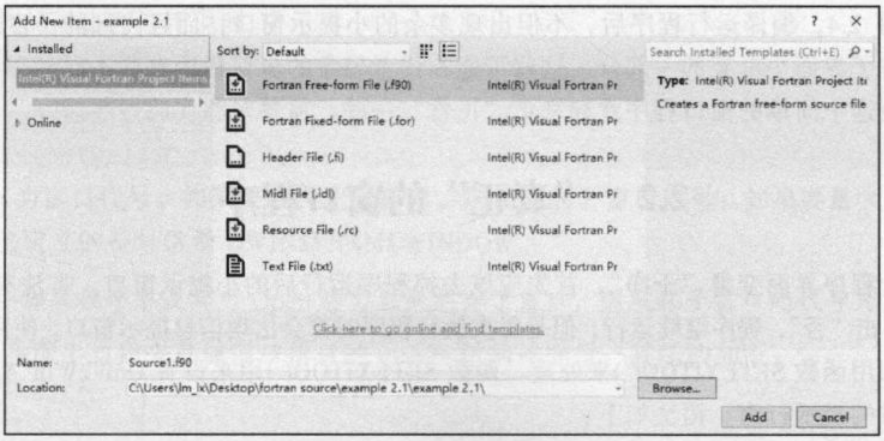

图 2.3　添加新项对话框

（6）选择 Fortran Free-form File(.f90)模板，单击 Add 按钮。

（7）在随后出现的源代码编辑器中，输入下面的代码。

【例 2.1】输入“hello world!”。

程序代码如下：

```
#001  ! example 2.1
#002  program main
#003  implicit none
#004  write(*,*)"hello world!"
#005  end program main
```

（8）按【Ctrl+F5】组合键运行程序，得到程序界面，如图 2.4 所示。

（9）程序编译运行后，随主窗口弹出一个小提示窗口。如果单击“否”按钮，然后最大化子窗口，则程序窗口界面如图 2.5 所示。

图 2.4　简单的窗口程序

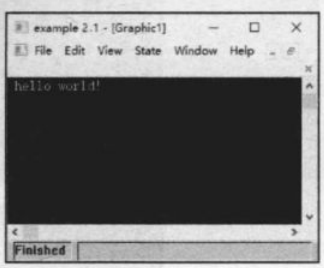

图 2.5　程序窗口界面

该简单示例中，程序代码只有 5 行，但是运行结果中出现了窗口程序界面及其典型的特征元素，如标题栏、菜单和状态栏等。

仅看代码，没有哪一句用来表述窗体相关元素的创建和显示。其实，QuickWin 工程中，编译器已经为用户做了很多工作，大大简化了窗口程序设计难度。

回看图 2.4，编译运行程序后，不但出现多余的小提示窗口，而且内部的子窗口（黑色部分）仅占了界面主框架的一部分。这些可能和读者想象的窗口程序有些不同。

为了使这个简单的窗口程序看起来更“正常”一些，需要对源代码做些修改。

2.2 “真正”的窗口程序

为了让程序界面变得“干净”，首先应该去掉程序运行后的小提示窗口。方法有 2 种：

（1）单击“否”，程序继续运行，但是每次执行程序时都会出现信息提示窗口，使用不方便。

（2）调用函数 SETEXITQQ( )来处理。函数 SETEXITQQ( )用来设置 QuickWin Application 程序退出时的提示行为。语法如下：

```
USE IFQWIN
integer( kind=4 )::exitmode, result
result=SETEXITQQ(exitmode)
exitmode = QWIN$EXITPROMPT          !为程序的默认情况
或        = QWIN$EXITNOPERSIST      !编译运行时，窗口出现后马上消失
或        = QWIN$EXITPERSIST        !运行后窗口正常显示，没有任何信息提示框
```

为了正确使用该函数，必须在程序代码中包含语句 USE IFQWIN。

【例 2.2】将函数 SETEXITQQ( )加入例 2.1。

程序代码如下：

```
#001 ! example 2.2
#002 program main
#003   USE IFQWIN
#004   implicit none
#005   integer(kind=4 )::results
#006   write(*,*) "hello world!"
#007   results=SETEXITQQ(QWIN$EXITPERSIST)
#008 end program main
```

编译并运行，结果如图 2.6 所示。

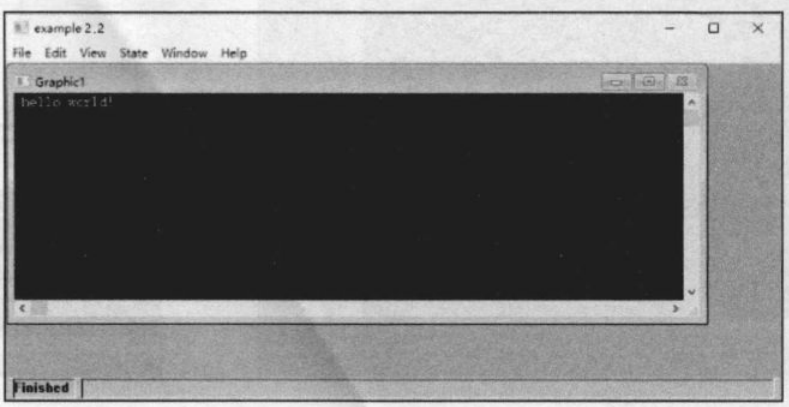

图 2.6　无信息提示的程序窗口

由例 2.2 可知，因为函数 SETEXITQQ( )的调用，程序运行后的小提示窗口不再出现。但是，图 2.6 中内部子窗口仅占有主界面的一部分，程序界面没有整体感。

如果程序主界面打开的同时，其子窗口能随着主窗口一起最大化，则将得到明显的整体观感。为此，需要调用函数 SETWSIZEQQ( )设置窗口（主窗口和子窗口）的尺寸和位置，下

面是该函数的语法和使用说明。

```
USE IFQWIN
integer( kind=4 )::result, unit
type(qwinfo)::winfo
result=SETWSIZEQQ(unit, winfo)
```

其中，unit 为窗口代号。如果要最大化子窗口，则 unit 为子窗口代号；如果要最大化主窗口，则其值为已定义的符号常量 QWIN$FRAMEWINDOW。

注意：如果程序中没有用 Open 语句显式打开子窗口，则默认子窗口的代号为 0、5 或 6。

winfo 是派生自用来存储窗口信息的 qwinfo 结构体变量，其结构如下：

```
TYPE QWINFO
  INTEGER(2) TYPE              ! 请求类型
  INTEGER(2) X                 ! 窗口左上角的 X 坐标
  INTEGER(2) Y                 ! 窗口左上角的 Y 坐标
  INTEGER(2) H                 ! 窗口高度
  INTEGER(2) W                 ! 窗口宽度
END TYPE QWINFO
```

其中，TYPE 可能的取值为：

Winfo% TYPE = QWIN$MIN 时，最小化窗口；

或 = QWIN$MAX 时，最大化窗口；

或 = QWIN$RESTORE 时，恢复最小化窗口到之前的大小；

或 = QWIN$SET 时，将根据 Winfo 中的其他值设置窗口的位置和大小。

【例 2.3】调用函数 SETWSIZEQQ( )，将主窗口和子窗口分别最大化。

程序代码如下：

```
#001 ! example 2.3
#002 program main
#003   USE IFQWIN
#004   implicit none
#005   integer(kind=4)::results
#006   type(qwinfo)::winfo
#007   winfo%type=QWIN$MAX
#008   ! 最大化主窗口
#009   results=SETWSIZEQQ(QWIN$FRAMEWINDOW, winfo)
#010   ! 最大化子窗口
#011   results=SETWSIZEQQ(0, winfo)
#012   write(*,*) "Hello World!"
#013   ! 取消提示信息
#014   results=SETEXITQQ(QWIN$EXITPERSIST)
#015 end program main
```

编译运行后，结果如图 2.7 所示。

图 2.7　窗口最大化

2.3　窗口和文本颜色

对多数工程技术人员来讲，数值计算本身有一定难度，再加上只有黑白两种呈现色，那么程序开发的体验显然是很糟糕的。为了呈现更好的计算环境和视觉效果，程序开发中颜色的设置是必要的。

QuickWin 工程中提供了多个用于颜色设置的函数。函数 SETBKCOLORRGB()用来设置窗口背景色。函数语法如下：

```
USE IFQWIN
integer(kind=4)::color, result
result=SETBKCOLORRGB(color)
```

其中，color 是背景色，有两种方式来指定：

（1）用十六进制数表达 RGB 分量，从左向右分别代表 B、G、R，如#FF0000 代表蓝色，#00FF00 代表绿色，#0000FF 代表红色。

（2）用函数 RGBTOINTEGER()将 RGB 分量值转换为代表相应颜色的整数。函数语法如下：

```
USE IFQWIN
integer(kind=4)::color
integer(kind=4)::red, green, blue
color=RGBTOINTEGER( red, green, blue )
```

【例 2.4】尝试改变窗口背景的颜色。

程序代码如下：

```
#001 ! example 2.4
#002 program main
#003   USE IFQWIN
#004   implicit none
#005   integer(kind=4 )::results, color
#006   type( qwinfo)::winfo
#007   color=RGBTOINTEGER(192, 192, 192)
#008   ! 改变背景颜色
#009   results=SETBKCOLORRGB(color)
#010   winfo%type=QWIN$MAX
#011   ! 最大化主窗口
#012   results=SETWSIZEQQ(QWIN$FRAMEWINDOW, winfo)
#013   ! 最大化子窗口
#014   results=SETWSIZEQQ(0, winfo)
#015   write(*,*) "hello world!"
#016   ! 取消提示信息
#017   results=SETEXITQQ(QWIN$EXITPERSIST)
#018 end program main
```

其中第 7～9 行的目的是设置窗口背景色。

按照我们的思路，设置背景颜色后，当程序运行时窗口背景颜色将变为第 7 行设置的灰色。但是，程序运行后，子窗口的背景色并没有发生变化。相比上例，变化的仅仅是文本的背景色，具体情况如图 2.8 所示。

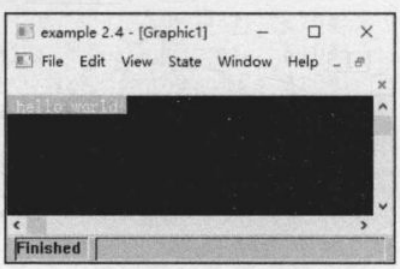

图 2.8　窗口背景色的设置结果

为解决这个问题，需要重新研究窗口背景色的设置函数 SETBKCOLORRGB()。原来，该函数只能设置窗口的背景色而不能改变窗口背景颜色。

为了改变背景颜色，需要调用子程序 CLEARSCREEN()。其语法如下：

```
USE IFQWIN
integer( kind=4 )::area
```