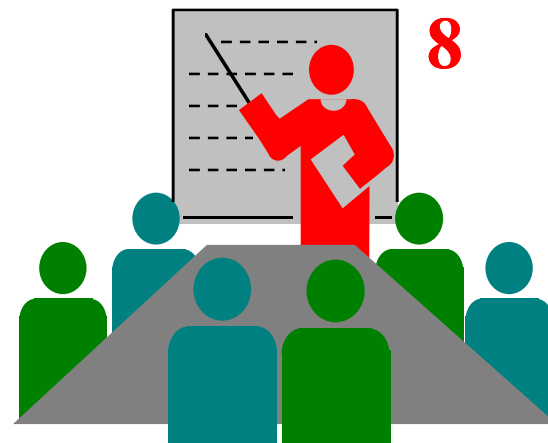


第二章 **FORTRAN90**开发环境

1. FORTRAN语言发展概述
2. FORTRAN90主要特征
3. Visual FORTRAN 6.6主要特性
4. Visual FORTRAN 6.6软硬件配置
5. Developer Studio开发环境概述
6. Developer Studio环境解题过程
7. MS-DOS命令行开发环境简介
8. 习题二



2.1 FORTRAN语言发展概述



- ◆ 语言地位
- ◆ 发展历程
- ◆ F90影响

FORTRAN是英文“**FOR**mula **TRAN**slator”的缩写,译为“**公式翻译器**”,它是世界上最早出现的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**。FORTRAN语言以其特有的功能在数值、科学和工程计算领域发挥着重要作用。

FORTRAN语言对程序设计语言的研究和发展有较大的影响。随着计算机科学技术的飞速发展,FORTRAN语言也不断得到完善和充实,从传统FORTRAN,到结构化FORTRAN,再到具有现代特征的FORTRAN90。



2.1 FORTRAN语言发展概述



- ◆ 语言地位
- ◆ 发展历程
- ◆ F90影响

- 1951年，美国IBM公司着手研究开发FORTRAN语言，并于1954年对外发布，称**FORTRAN I**。
- 1958年，扩充和完善FORTRAN I，引入函数概念，投入商业运行，推出**FORTRAN II**。
- 1960年，推出**FORTRAN III**，但没有在计算机上实现。
- 1962年，推出了**FORTRAN IV**。没有充分考虑兼容性。
- 1966年，推出**FORTRAN 66**，美国标准，后被ISO采用。
- 1978年，推出**FORTRAN 77**，兼容性好，结构化特征。
- 1991年，推出**FORTRAN 90**，现代语言，可视化编程。
- 1995年，微软公司在Developer Studio集成开发平台上推出**FORTRAN PowerStation 4.0**。引入可视化编程技术。
- 1997年，微软和数据设备公司合作推出**Digital Visual FORTRAN 5.0**，功能强大。
- 1998年，康柏公司与数据设备公司合并，推出**Compag Visual FORTRAN 6.x**。



2.1 FORTRAN语言发展概述



- ◆ 语言地位
- ◆ 发展历程
- ◆ F90影响

FORTRAN 90的推出，是FORTRAN语言发展史上的一个里程碑，它挽救了FORTRAN语言面临淘汰的局面，使传统的FORTRAN语言焕发了青春，以其强大的功能和严谨的风格受到人们的欢迎和认可，并展现出美好的应用前景。

西方发达国家已开始重新认识FORTRAN语言，对FORTRAN90语言的研究、开发和应用给予了高度重视，FORTRAN语言正在发挥越来越大的作用。

我国最近两年开始引进FORTRAN90有关资料和相关软件，许多高校开始开设FORTRAN90课，为提高学生程序设计水平和能力打下了坚实基础。



2.2 FORTRAN90主要特征



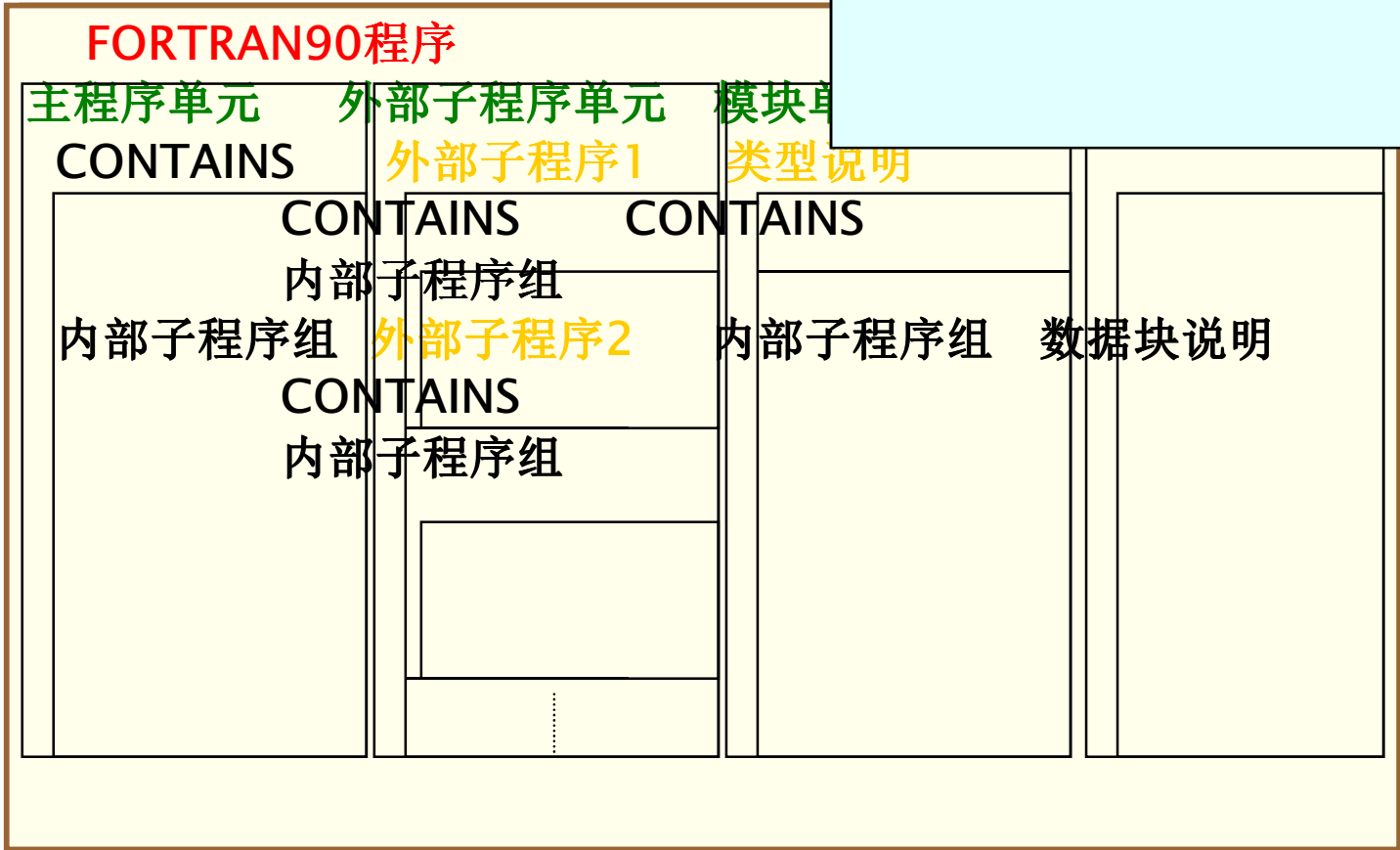
- ◆ 组织结构
- ◆ 书写形式
- ◆ 结构化
- ◆ 过程
- ◆ 作用域
- ◆ 关联
- ◆ 数据类型
- ◆ 数组
- ◆ 指针
- ◆ 输入输出
- ◆ 语法检查


- FORTRAN 90采用模块化和结构化相结合的程序组织结构。
- FORTRAN 90程序由四部分组成：主程序单元、外部子程序单元、模块单元和数据块单元。
- 程序单元可分别保存在独立的源程序文件中,分别编译,最后连接和运行。
- FORTRAN 90程序有且只能有一个主程序单元,其它程序单元可有可无,可多可少。
- 子程序分内部子程序和外部子程序,有不同的使用范围,外部子程序只能在外部子程序单元中说明和定义,内部子程序只能在主程序、外部子程序模块中的CONTAINS和END语句之间说明和定义。模块单元中的子程序称为模块子程序。

程序组织结构如图2-1所示。



```
PROGRAM main
  READ*,a,b
  PRINT *,add(a,b)
CONTAINS
  FUNCTION add(a,b)
    add=a+b
  END FUNCTION
END
```





```
PROGRAM
main
READ*,a,b,c
s=add(a,b,c)
PRINT *,s
END
```

```
FUNCTION add(a,b,c)
  add=f(a,(f(b,c))
CONTAINS
FUNCTION f(x,y)
  f=x+y
END FUNCTION
END FUNCTION
```

2.2 FORTRAN90主要特征



- ◆ 组织结构
- ◆ 书写形式
- ◆ 结构化
- ◆ 过程
- ◆ 作用域
- ◆ 关联
- ◆ 数据类型
- ◆ 数组
- ◆ 指针
- ◆ 输入输出
- ◆ 语法检查

- FORTRAN 90源程序有自由格式和固定格式两种书写格式。
- 固定格式采用FORTRAN 77书写格式,有严格的书写限制。固定格式规定:一个程序单元由若干行语句组成,每行80个字符,分四个区域:标号区、续行区、语句区和注释区,如图2-2所示。
- 自由格式采用PASCAL、C等现代语言书写格式,书写不再受分区和位置限制。自由格式规定:一个程序单元由若干行语句组成,一行长度可任意(默认为132个字符);“!”字符后均为注释说明文字;标号与语句之间用空格间隔;一行可写多条语句,语句之间用“;”字符间隔;语句可从任意位置开始书写,如图2-3所示。



固定格式示例

标号区 续行区
(1-5列) (6列)

语句区
(7-72列)

注释区
(73-80列)

!	求一元二次方程两个实根	
	PROGRAM quadratic	
	REAL a,b,c,d,x1,x2	
	READ *,a,b,c	输入系数
	d=b*b-4*a*c	算判别式
100	IF (d>=0) THEN	
	x1=(-b+SQRT(d))/(2*a)	
	x2=(-b-SQRT(d))	
&	/(2*a)	
	PRINT *, 'x1=', x1	
	PRINT *, 'x2=', x2	
	ELSE	
	PRINT *, '该方程有复数根。'	
	END IF	
	END	程序结束

自由格式示例

! 求一元二次方程两个实根	
PROGRAM quadratic	
REAL a,b,c,d,x1,x2	
READ *,a,b,c !输入系数	
d=b*b-4*a*c !计算判别式	
IF (d>=0) THEN	
x1=(-b+SQRT(d))/(2*a); x2=(-b-SQRT(d))/(2*a)	
PRINT *, 'x1=' , x1	
PRINT *, 'x2=' , x2	
ELSE	
PRINT *, '该方程有复数根。'	
END IF	
END !程序结束	

2.2 FORTRAN90主要特征



- ◆ 组织结构
- ◆ 书写形式
- ◆ 结构化
- ◆ 过程
- ◆ 作用域
- ◆ 关联
- ◆ 数据类型
- ◆ 数组
- ◆ 指针
- ◆ 输入输出
- ◆ 语法检查

● FORTRAN 90提供了块IF、块CASE、DO循环和DO WHILE循环等结构化控制语句，都将END IF、END SELECT或END DO作为语句的结束子句，取消了CONTINUE语句。FORTRAN 90还允许对结构化控制语句进行命名，如图2-4所示。

```
Sum: DO I<=10  
      S=S+I  
      I=I+1  
END DO Sum
```



2.2 FORTRAN90主要特征



- ◆ 组织结构
- ◆ 书写形式
- ◆ 结构化
- ◆ 过程
- ◆ 作用域
- ◆ 关联
- ◆ 数据类型
- ◆ 数组
- ◆ 指针
- ◆ 输入输出
- ◆ 语法检查

- 过程是被命名的一组语句,也称子程序,通过调用执行,完成一系列运算或操作,获得运行结果。
- 过程分函数和子程序。
- FORTRAN 90增加了递归过程。

```
PROGRAM main
  READ *,a,b
  c=add(a,b)
  PRINT *,c
CONTAINS
FUNCTION
add(x,y)
  add=x+y
END    FUNCTION
add
END
```



2.2 FORTRAN90主要特征



- ◆ 组织结构
- ◆ 书写形式
- ◆ 结构化
- ◆ 过程
- ◆ 作用域
- ◆ 关联
- ◆ 数据类型
- ◆ 数组
- ◆ 指针
- ◆ 输入输出
- ◆ 语法检查

- 作用域是指程序中名称(变量名、子程序名、函数名等)的作用范围,FORTRAN 90规定了名称的作用域。
- 主程序单元中声明的名称其作用域为整个程序以及与公用块有关的所有外部子程序。
- 子程序内声明的名称其作用域为子程序本身。
- 不同作用域声明的名称可以重名,只在自身作用域内起作用,互不干扰。

作用域说明如图2-5所示。



作用域示例

```
PROGRAM main  
REAL a,b,c
```

```
CONTAINS
```

```
  SUBROUTINE p1(...)  
    REAL b',c',d
```

```
  END SUBROUTINE p1
```

```
  SUBROUTINE p2(...)  
    REAL c'',d',e
```

```
  END SUBROUTINE p2
```

```
END
```

b',c',d
的作用域

c'',d',e
的作用域

a,b,c
的作用域

2.2 FORTRAN90主要特征



- ◆ 组织结构
- ◆ 书写形式
- ◆ 结构化
- ◆ 过程
- ◆ 作用域
- ◆ 关联
- ◆ 数据类型
- ◆ 数组
- ◆ 指针
- ◆ 输入输出
- ◆ 语法检查

■ ➔ 关联是将同一作用域单元的变量联系起来,允许已声明的变量可被同一作用域内的其它变量访问。

有三种关联: **名称关联、指针关联和存储关联。**



2.2 FORTRAN90主要特征



- ◆ 组织结构
- ◆ 书写形式
- ◆ 结构化
- ◆ 过程
- ◆ 作用域
- ◆ 关联
- ◆ 数据类型
- ◆ 数组
- ◆ 指针
- ◆ 输入输出
- ◆ 语法检查

FORTRAN 90提供五种预定义的内部数据类型(**INTEGER**、**REAL**、**COMPLEX**、**CHARACTER**、**LOGICAL**),并引入了派生数据类型特征,提供了派生类型声明语句TYPE...ENDTYPE,使FORTRAN 90能说明和定义复杂的数据结构。

FORTRAN 90同时对五种预定义的数据类型进行了扩充,为其指定了一个类型参数值(KIND值),用于指定数据类型在内存中的方式和取值范围。



2.2 FORTRAN90主要特征



- ◆ 组织结构
- ◆ 书写形式
- ◆ 结构化
- ◆ 过程
- ◆ 作用域
- ◆ 关联
- ◆ 数据类型
- ◆ 数组
- ◆ 指针
- ◆ 输入输出
- ◆ 语法检查

数组是一组有序的、数据类型相同的数据集合。

→ FORTRAN 90对数组的操作有了很大改进,提供了许多能直接对数组进行操作的语句和函数,数组可按单个数据对象进行处理。



2.2 FORTRAN90主要特征



- ◆ 组织结构
- ◆ 书写形式
- ◆ 结构化
- ◆ 过程
- ◆ 作用域
- ◆ 关联
- ◆ 数据类型
- ◆ 数组
- ◆ 指针
- ◆ 输入输出
- ◆ 语法检查

指针是具有动态属性的变量,用于构建动态数据结构的一种数据形式。

FORTRAN 90引入了指针概念,新增了指针类型数据,使数据的动态构建和访问成为可能。



2.2 FORTRAN90主要特征



- ◆ 组织结构
- ◆ 书写形式
- ◆ 结构化
- ◆ 过程
- ◆ 作用域
- ◆ 关联
- ◆ 数据类型
- ◆ 数组
- ◆ 指针
- ◆ 输入输出
- ◆ 语法检查

FORTRAN 90扩充了输入输出功能,引入了新的输入输出方法,可设定文件操作的访问权限,可了解文件的开关状态。
READ和WRITE语句功能得到了扩充。



2.2 FORTRAN90主要特征



- ◆ 组织结构
- ◆ 书写形式
- ◆ 结构化
- ◆ 过程
- ◆ 作用域
- ◆ 关联
- ◆ 数据类型
- ◆ 数组
- ◆ 指针
- ◆ 输入输出
- ◆ 语法检查 →

FORTRAN 90强化了程序的语法检测功能,能对语法进行有效检测并加以报告,编程人员可通过编译开关决定是否对检测结果进行报告。



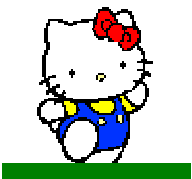
2.3 Visual FORTRAN 5.0主要特性



- ◆ 概述 ■ →
- ◆ 版本
- ◆ 特性

Digital Visual FORTRAN 5.0是微软公司Microsoft FORTRAN PowerStation 4.0的升级换代产品,它是微软和DEC公司于1997年起合作开发的产品,是一个集编译、连接、调试和运行于一身的FORTRAN 90集成开发环境。

Digital Visual FORTRAN 5.0是一个可视化编程工具。



2.3 Visual FORTRAN 5.0主要特性



- ◆ 概述
- ◆ 版本
- ◆ 特性

Digital Visual FORTRAN 5.0有两个版本:

- 标准版包括:

- ◆ Digital Visual FORTRAN(DVF)编译器
- ◆ DVF程序库
- ◆ Developer Studio

- 专业版包括:

- ◆ DVF编译器
- ◆ DVF程序库
- ◆ Developer Studio
- ◆ IMSL程序库



2.3 Visual FORTRAN 5.0主要特性



概述 版本 特性

- 支持FORTRAN 90标准, 支持多种平台FORTRAN 语言扩展。
- 可共享Windows资源和微软产品。软件开发速度快、效率高。
- 支持COM组件对象模型(Component Object Model)和OLE自动对象(OLE Automation Objects), 提供模块向导功能, 可快速产生调用COM和OLE对象的FORTRAN 90模块。
- 专业版提供了处理数学和统计数据的IMSL数值库, 库中给出了源程序和模块文件、在线帮助、在线README文件、大量应用例子以及用于多处理和高性能系统的算法。
- 支持命令行界面(文本、DOS), 允许定制命令行窗口环境。
- 兼容性好, 提供多种平台和多种版本扩展特性。
- 提供强大的在线帮助系统InfoViewer, 可实时获得帮助信息。
- 支持多语言混合编程。可混合编程的语言有: Visual FORTRAN、Visual C++、Visual J++、Visual Basic和Microsoft MASM。
- 微软公司Visual 系列产品之一, 可与其它Visual产品协作。



2.4 Visual FORTRAN 5.0配置与安装



- ◆ 硬件配置
- ◆ 软件配置
- ◆ 系统安装

- 要求Intel 486/66 MHz以上CPU,或完全兼容CPU。
- 对于Windows95/98/2000下的x86系统,要求16M以上内存;对于Windows NT下的x86系统,要求24M以上内存;对于Windows NT下的Alpha系统,要求32M以上内存。
- 要求32速以上光驱。
- 对于标准版(x86系统),要求30M(光驱运行安装)~190M(完全安装)硬盘;对于专业版(x86系统),要求30M~240M;对于专业版(Alpha系统),要求40M~260M。
- 要求VGA以上显示器,推荐17英寸SVGA显示器。
- 鼠标或其它兼容定位设备。



2.4 Visual FORTRAN 5.0配置与安装



- ◆ 硬件配置
- ◆ 软件配置
- ◆ 系统安装

- Windows 95/98/2000/NT 4.0,或UNIX、OpenVMS操作系统平台(必须)。
- Visual C++、Visual J++、Visual Basic、MASM程序设计语言(可选)。
- Microsoft Excel 97/2000数据图示工具(可选)。
- Visual KAP优化与并行处理(可选),用于优化和并行处理FORTRAN代码。
- Internet Explorer网络浏览软件(可选),用于浏览软件在线信息。
- 微软公司的其它软件产品(可选)。



2.4 Visual FORTRAN 5.0配置与安装



- ◆ 硬件配置
- ◆ 软件配置
- ◆ 系统安装

- 启动Windows系统平台;
- 若系统平台为Windows NT,则以受权名称登录;
- 把Digital Visual FORTRAN 5.0光盘插入光驱;
- 打开光盘,双击执行SETUP程序,弹出安装主窗口。
- 在主窗口中,单击“Install Visual FORTRAN”命令按钮;
- 弹出“Registration”对话框,输入注册信息(名称、单位、注册号),单击“Next”按钮,继续安装;
- 弹出“Setup Install Type”对话框,给出三种安装类型:典型安装(Typical)、定制安装(Custom)和由光驱运行安装(Run From CD-ROM)。选择安装类型,单击“Next”按钮,继续安装;
- 弹出“Select Folder”对话框,指定程序组名,缺省为“Visual FORTRAN 5.0”,单击“Next”按钮,复制文件,创建程序组和图标。
- 弹出“Setup Complete”对话框,完成安装;



2.5 Developer Studio开发环境概述



- ◆ 简介
- ◆ 工作空间
- ◆ 窗口说明

- Microsoft Developer Studio是微软公司开发的一个可视化集成开发环境(IDE),为多种软件系统提供统一的可视化集成开发平台。微软许多软件产品共享该平台。
- 软件开发环境,特别是IDE的研究、开发和应用,长期受到计算机厂商和一些研究机构的关注。推出若干软件开发环境,如: Turbo、IDE等。
- 90年代,微软推出可视化编程工具和可视化软件开发环境,如Visual系列编程工具。Microsoft Developer Studio是这些编程工具统一使用的共享平台。
- 微软将IDE引入FORTRAN 90, 推出Microsoft FORTRAN PowerStation 4.0和Digital Visual FORTRAN 5.0,开发环境与FORTRAN 90形成一个有机整体。
- Microsoft Developer Studio将文本编辑器、资源编辑器、项目创建工具、增量连接器、源程序浏览器、程序调试器和信息查询器等集成在一起,以可视化形式进行程序的编辑、编译、调试、运行等操作,用户在统一的视窗界面上操作,完成FORTRAN 90程序的设计和开发。



2.5 Developer Studio开发环境概述



- ◆ 简介
- ◆ 工作空间
- ◆ 窗口说明

- 用户在开发软件过程中会创建和产生大量的源程序文件、资源文件和其它类型辅助文件(需求分析、规格说明书、流程图、开发计划等),这些文件相互关联,合理地组织这些文件是软件开发的一项重要工作,Windows中的“我的电脑”资源管理器就是合理组织文件的有效工具。
- Developer Studio中的工作空间和项目功能类似“我的电脑”资源管理器功能。通过工作空间和项目来合理地组织文件。
- 工作空间: 项目集合。一个工作空间创建一个文件夹,且文件夹内包含若干项目(子文件夹)和二个管理这些项目的数据库文件(.opt,.dsw)。
- 项目: 与软件有关的源程序文件、资源文件、辅助文件的集合。一个项目构成可独立运行的软件。一个项目创建一个文件夹,且文件夹内包含源程序文件、资源文件、辅助文件、可执行文件、数据库管理文件。
- 用户、工作空间、项目、文件之关系。(示例)



2.5 Developer Studio开发环境概述



- ◆ 简介
- ◆ 工作空间
- ◆ 窗口说明

主窗口：标题条、菜单条、工具条、状态条和环境窗口。

- **标题条：**窗口顶部,给出当前工作空间名和文件名。
- **菜单条：**标题条下方,给出常用功能的菜单操作。
- **工具条：**给出常用操作,有9个预定义工具条。
- **状态条：**窗口底部,显示操作说明、行列号、时间等信息
- **环境窗口：**提供软件开发的工作场所。有二种窗口：停靠窗口和文档窗口。

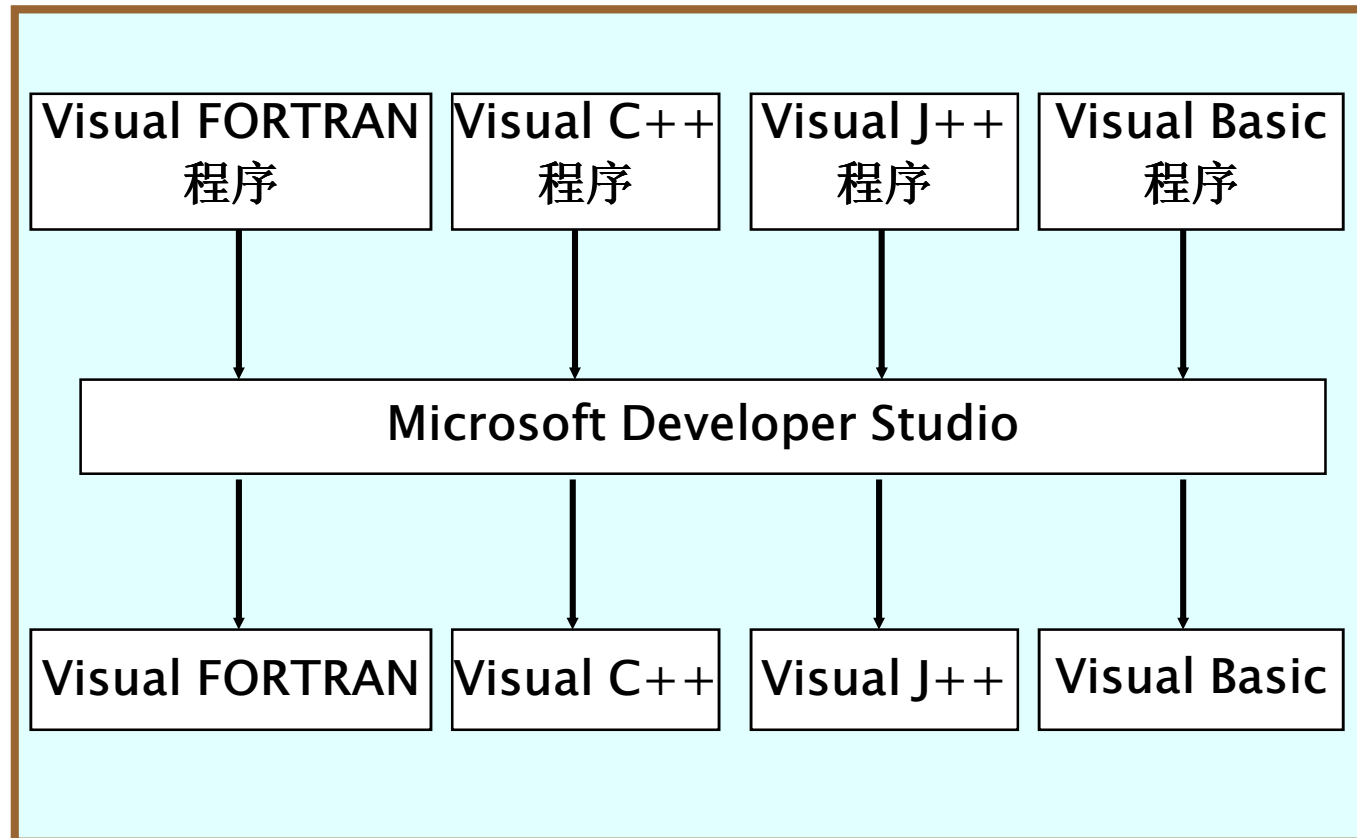
◆ **停靠窗口：**可拖至主窗口顶、底、左、右部位(固定窗口),也可拖至主窗口中央(浮动窗口)。固定窗口和浮动窗口可相互切换。两个常用停靠窗口：工作空间窗口和输出窗口。

◆ **文档窗口：**位于停靠窗口一侧,最大化显示只有边框而无标题,不能拖动。两个常用文档窗口：源程序文档窗口和信息查询文档窗口。

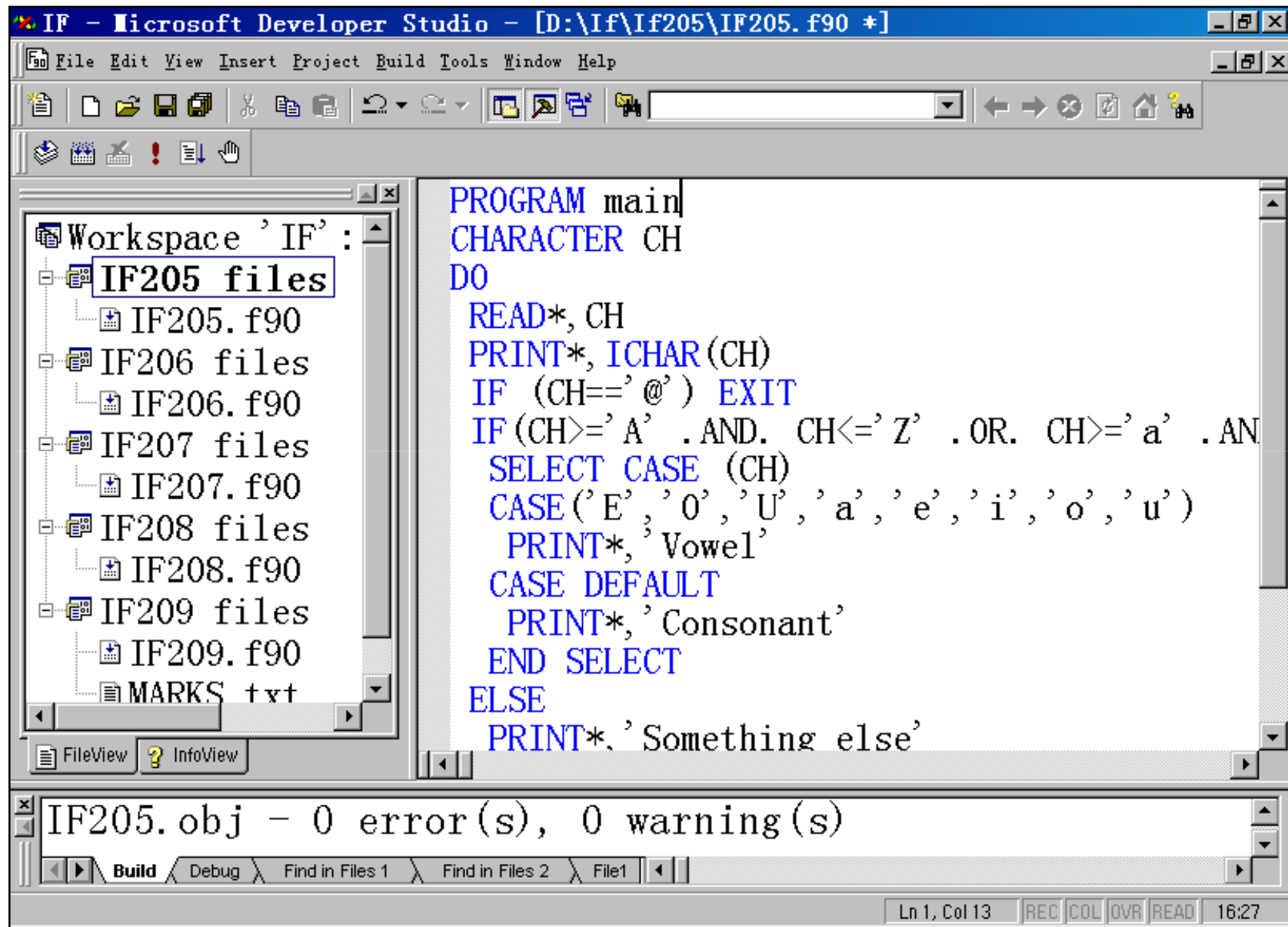
菜单条、工具条、停靠窗口都有停靠和浮动特性。



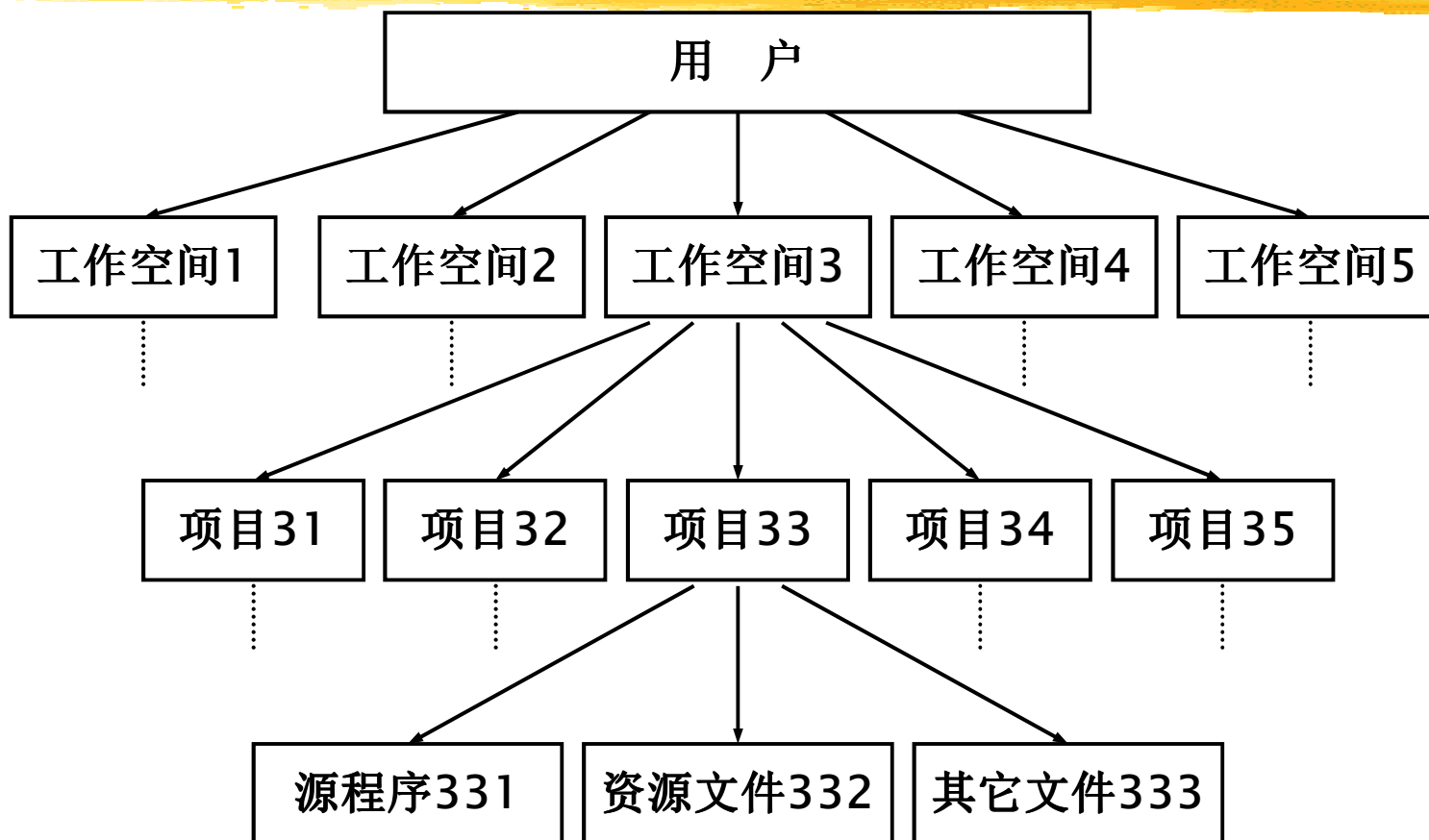
编程工具与开发环境关系



开发环境主窗口

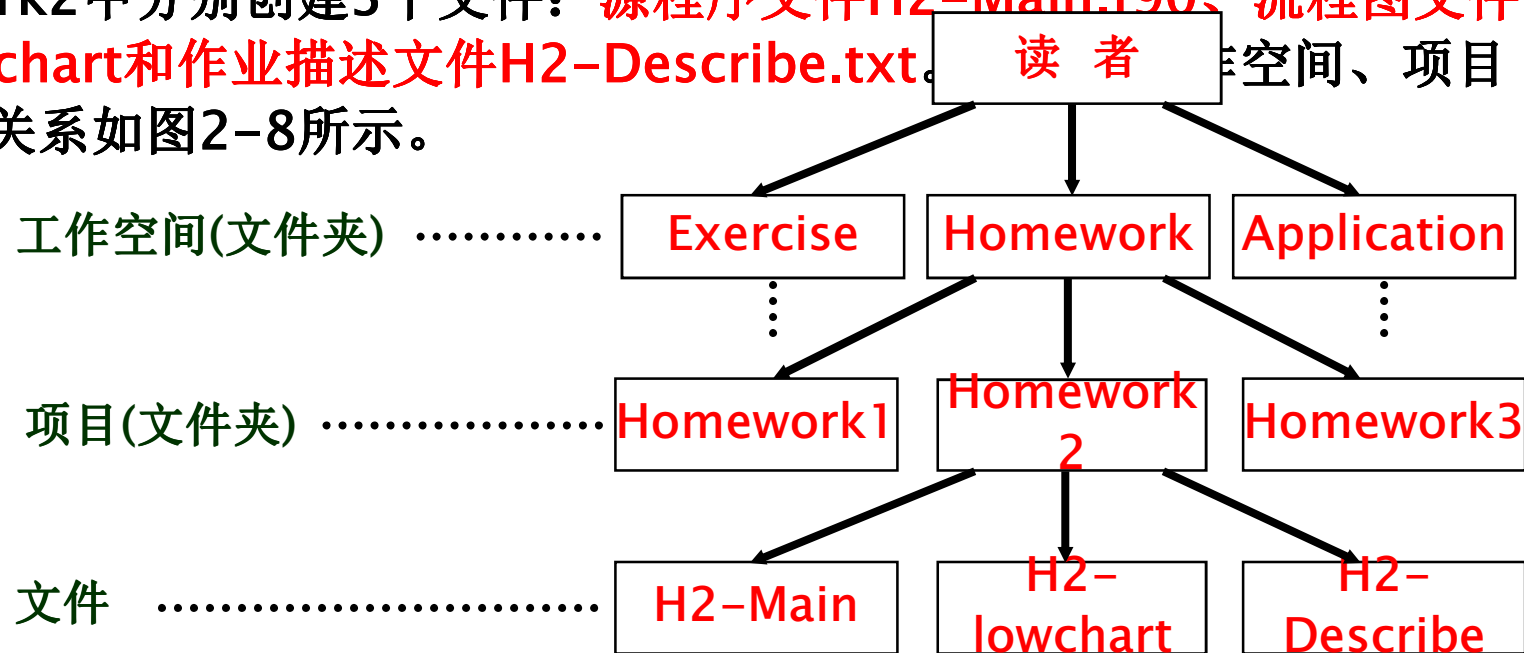


工作空间、项目、文件之关系

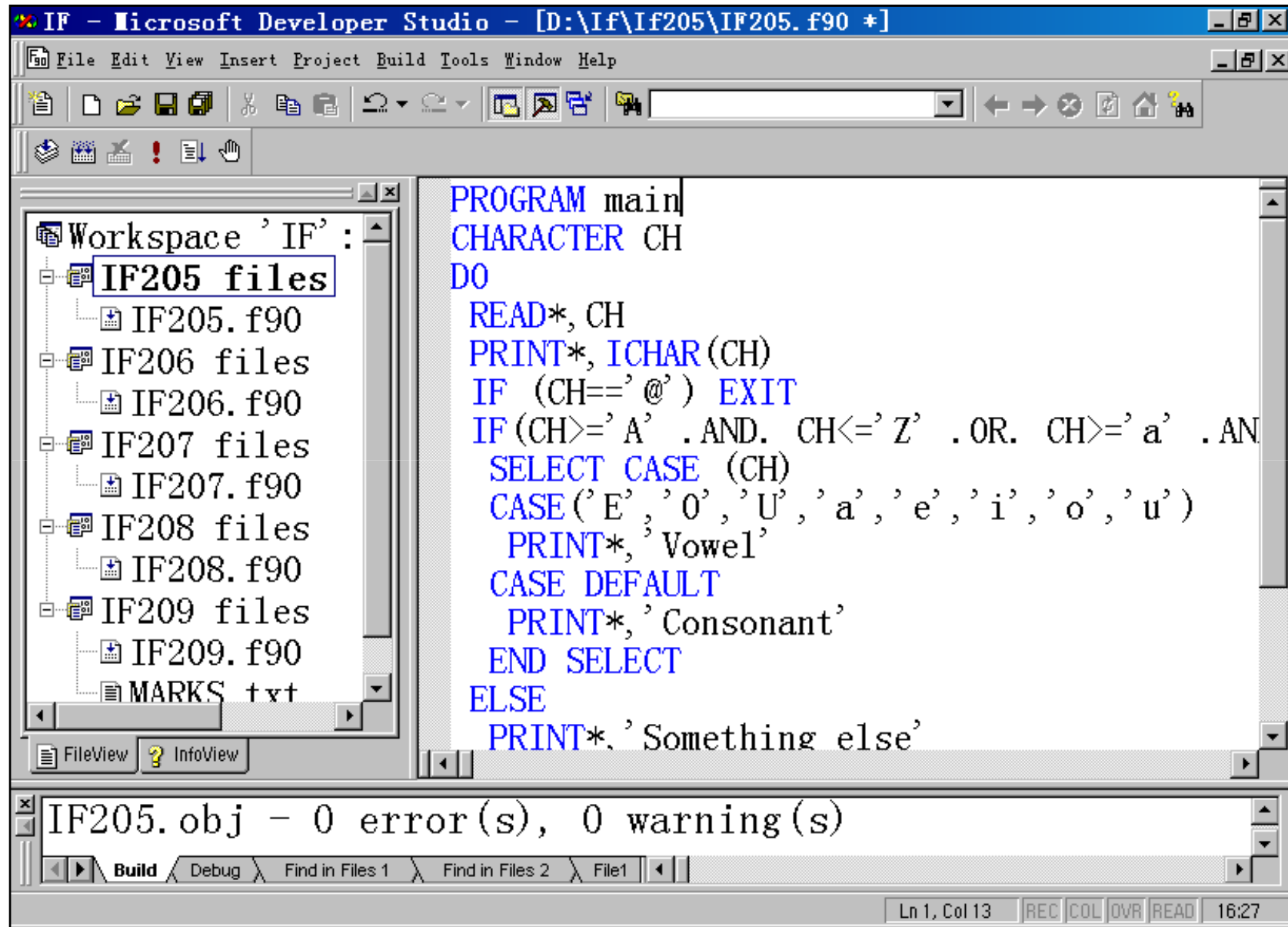


工作空间、项目、文件之关系示例

学生在学习FORTRAN 90过程中,设计和编写3种FORTRAN程序: **一是作业程序;二是练习程序;三是应用程序**。可分别创建3个工作空间:**Homework、Exercise,Application**,分别组织作业程序、练习程序和应用程序,以及相关文件。如果有3个作业: Homework1、Homework2和Homework3,则在Homework工作空间中分别创建3个项目: **Homework1、Homework2和Homework3**,分别组织这些作业的作业程序及相关文件。如果作业Homework2有源程序、流程图和作业描述等文件,则在项目Homework2中分别创建3个文件: **源程序文件H2-Main.f90、流程图文件H2-Flowchart和作业描述文件H2-Describe.txt**。



开发环境主窗口说明



2.6 Developer Studio解题过程



- ◆ 概述
- ◆ 前期准备
- ◆ 创建工作空间
- ◆ 创建项目
- ◆ 创建源程序文件
- ◆ 创建辅助文档
- ◆ 编译源程序
- ◆ 构建可执行程序
- ◆ 运行可执行程序
- ◆ 调试程序
- ◆ 程序多区域显示

◆用户在Developer Studio上主要完成以下工作:

- 分析问题, 设计算法, 绘制框图, 编写程序。
- 创建有关工作空间、项目(已建, 可省略)。
- 创建并输入源程序, 创建其它辅助文档。
- 编译源程序。
- 构建、连接生成目标程序(可执行程序)。
- 运行目标程序。
- 调试程序。

通过实例简要说明解题过程。

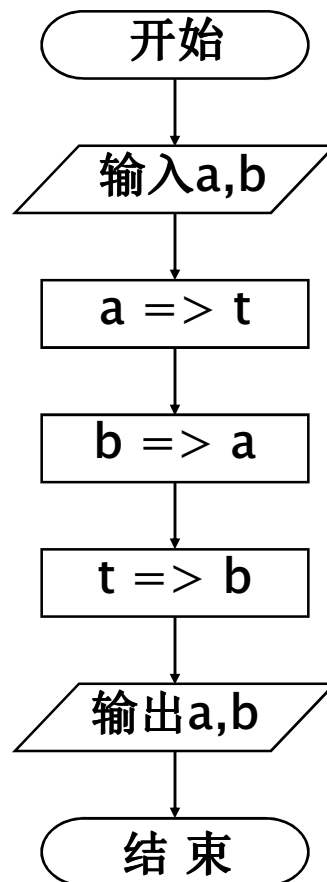
- 问题一: 输入两个整数, 对其进行交换, 将交换后结果从屏幕上输出。
- 问题二: 输入三个整数, 对其进行从小到大排序, 将排序结果从屏幕上输出。

2.6 Developer Studio解题过程



- ◆ 概述
- ◆ 前期准备
- ◆ 创建工作空间
- ◆ 创建项目
- ◆ 创建源程序文件
- ◆ 创建辅助文档
- ◆ 编译源程序
- ◆ 构建可执行程序
- ◆ 运行可执行程序
- ◆ 调试程序
- ◆ 程序多区域显示

用户在上机操作前应做好充分的前期准备工作，认真分析问题、精心设计算法，详细绘制框图，准确编写程序。以下是问题一的框图和程序。



! 两整数交换

PROGRAM exchange

INTEGER a,b,t

READ *,a,b

t=a

a=b

b=t

PRINT *,a,b

END

2.6 Developer Studio解题过程



- ◆ 概述
- ◆ 前期准备
- ◆ 创建工作空间
- ◆ 创建项目
- ◆ 创建源程序文件
- ◆ 创建辅助文档
- ◆ 编译源程序
- ◆ 构建可执行程序
- ◆ 运行可执行程序
- ◆ 调试程序
- ◆ 程序多区域显示

工作空间名指定为：**example**，在D盘上创建一个工作空间：**example**。

创建步骤：

- 单击选择执行“File”/“New”菜单项，弹出“New”对话框，选取“Workspaces”选项卡，如图2-22所示。
- 在名称和位置框分别键入工作空间名和路径。路径输入可通过浏览窗口查找和定位。
- 单击“OK”按钮创建新的工作空间。

在工作空间窗口内建立新的选项卡：FileView，同时显示“Workspace'example':0 Project(s)”，指出工作空间名称和项目个数，如图2-23所示。在D盘上创建新的文件夹：**D:\example**，并生成两个工作空间管理文件：**example.opt**和**example.dsw**。要打开工作空间:example,打开example.dsw即可。

(操作演示)

2.6 Developer Studio解题过程



- ◆ 概述
- ◆ 前期准备
- ◆ 创建工作空间
- ◆ 创建项目
- ◆ 创建源程序文件
- ◆ 创建辅助文档
- ◆ 编译源程序
- ◆ 构建可执行程序
- ◆ 运行可执行程序
- ◆ 调试程序
- ◆ 程序多区域显示

问题一项目名指定为：**example1**，在工作空间**example**中针对问题一创建一个项目：**example1**。

创建步骤：

- 打开“New”对话框,选取“Projects”选项卡,如图2-24所示;
- 在文本框分别键入项目名称和路径,路径输入也可单击右侧按钮打开浏览窗口查找和定位路径。单击“Add to current workspace”选项,选择应用程序类型: Win32 Console Application, 即控制台应用程序。指定运行平台;
- 完成输入后,单击“OK”按钮创建新的项目。

在工作空间**example**内建立新项目：**example1**,同时创建子文件夹：**example1**，在项目文件夹**example1**中生成一个项目管理文件：**example1.dsp**。

同法,针对问题二创建新的项目：**example2**。

(操作演示)

2.6 Developer Studio解题过程



- ◆ 概述
- ◆ 前期准备
- ◆ 创建工作空间
- ◆ 创建项目
- ◆ 创建源程序文件 →
- ◆ 创建辅助文档
- ◆ 编译源程序
- ◆ 构建可执行程序
- ◆ 运行可执行程序
- ◆ 调试程序
- ◆ 程序多区域显示

在项目中创建源程序文件,根据需要创建其它资源文件和辅助文档。针对问题一在项目 **example1** 中创建源程序文件: **example1.f90**。

创建步骤:

- 打开 “New”对话框,选取 “Files”选项卡;
- 选取 “Add to project”选项,选择项目名,在文本框键入源程序文件名称和路径。选择自由格式;
- 完成输入后,单击 “OK”按钮创建新的源程序文件。

创建文件后,FileView选项卡中项目: example1下,建立新的源程序文件名: **example1.f90**,同时在文件夹: D:\example\example1中生成该文件。在右侧打开一空白源程序文档窗口,在文档窗口中编辑输入源程序,如图2-28所示。以后若需打开源程序,则打开这个文件即可。

(操作演示)

2.6 Developer Studio解题过程



- ◆ 概述
- ◆ 前期准备
- ◆ 创建工作空间
- ◆ 创建项目
- ◆ 创建源程序文件
- ◆ 创建辅助文档
- ◆ 编译源程序
- ◆ 构建可执行程序
- ◆ 运行可执行程序
- ◆ 调试程序
- ◆ 程序多区域显示

有时需创建相关的资源文件或辅助文件。资源文件有：位图文件、光标文件、图标文件等。辅助文件有：问题描述文件、规格说明文件、流程图文件等。创建这些文件可在Developer Studio中直接运行有关软件完成。针对问题一创建两个辅助文档文件：**flowchart-exam1.doc**和**problem-exam1.txt**。

创建流程图文档文件步骤：

- 打开“New”对话框,选取“Other Documents”选项卡,如图2-29所示;
- 选取“Add to project”选项,选择项目名,键入文件名称和路径。选择Word文档类型;
- 完成输入后,单击“OK”按钮创建新的文档文件。

在FileView选项卡中项目：example1下,建立新的辅助文档文件,同时在项目文件夹中生成该文件。在右侧打开一空白Word文档窗口,在文档窗口中编辑、输入和绘制流程图,工具条被Word软件工具条替换。

(操作演示)

2.6 Developer Studio解题过程



- ◆ 概述
- ◆ 前期准备
- ◆ 创建工作空间
- ◆ 创建项目
- ◆ 创建源程序文件
- ◆ 创建辅助文档
- ◆ 编译源程序
- ◆ 构建可执行程序
- ◆ 运行可执行程序
- ◆ 调试程序
- ◆ 程序多区域显示

需要对源程序进行编译,生成中间程序文件(扩展名为.obj)和数据库文件(扩展名为.pdb),以便连接和运行。编译过程中检查、发现和排除语法错误。

1. 编译前参数设置

编译前根据需要设置有关参数,如:obj文件和pdb文件存放位置(一般取默认debug)。obj文件是编译后产生的文件,pdb文件是存放编译信息的数据库文件。

通过“Settings”菜单项,“General”选项卡设置。

2. 编译

3. 语法检查

在编译过程中,进行有关语法检查。编译结束后,在输出停靠窗口显示编译结果信息。如果无语法错误,则显示信息“example1.obj - 0 error(s), 0 warning(s)”,否则显示错误提示信息。若不小心把语句“READ *,a,b”错误写成“REAF *,a,b”,则显示语法错误信息。(操作演示)

2.6 Developer Studio解题过程



- ◆ 概述
- ◆ 前期准备
- ◆ 创建工作空间
- ◆ 创建项目
- ◆ 创建源程序文件
- ◆ 创建辅助文档
- ◆ 编译源程序
- ◆ 构建可执行程序
- ◆ 运行可执行程序
- ◆ 调试程序
- ◆ 程序多区域显示

obj文件是不能直接执行,必须构建生成可执行程序文件(扩展名为: .exe)才能使程序在计算机上运行并获得运行结果。

所谓程序构建(也称连编)就是将obj文件与系统提供的有关环境参数、预定义子程序和预定义函数等连接在一起,生成完整的可执行程序代码。构建过程中也能检查、发现和排除一些语法错误。

1. 构建前参数设置

构建前可设置有关参数,如:exe文件存放位置。通过“Settings”菜单项,“Link”选项卡设置。

2. 构建

3. 语法检查

在构建过程中,进行错误检查。在输出窗口显示构建结果信息。若无错误,则显示“example1.exe - 0 error(s), 0 warning(s)”,否则显示错误提示信息。信息说明形式类似编译信息说明。(操作演示)

2.6 Developer Studio解题过程



- ◆ 概述
- ◆ 前期准备
- ◆ 创建工作空间
- ◆ 创建项目
- ◆ 创建源程序文件
- ◆ 创建辅助文档
- ◆ 编译源程序
- ◆ 构建可执行程序
- ◆ 运行可执行程序
- ◆ 调试程序
- ◆ 程序多区域显示

产生可执行程序(exe文件)后,需运行可执行程序才能获得运行结果。运行过程中也能检查、发现和排除一些语法错误。

运行程序步骤:

- 通过“Build”下拉菜单运行程序。
- 通过“开始”菜单的“运行”菜单项运行程序。
- 通过Windows系统的资源管理器运行程序。
- 通过MS-DOS方式运行程序。

▶ 程序中如果有输入输出操作语句,则运行时出现输入输出窗口,在窗口中输入数据。

运行过程中,若出现运行错误,则在输出窗口显示运行错误信息。如:执行除法时遇到分母为0,将出现运行错误。用户改正有关错误,重新编译、构建和运行,直到运行正确。



- ◆ 概述
- ◆ 前期准备
- ◆ 创建工作空间
- ◆ 创建项目
- ◆ 创建源程序文件
- ◆ 创建辅助文档
- ◆ 编译源程序
- ◆ 构建可执行程序
- ◆ 运行可执行程序
- ◆ 调试程序
- ◆ 程序多区域显示

要得到一个无错误(语法错误、运行错误和逻辑错误)的可执行程序并非易事,需要有扎实的程序设计知识、丰富的程序设计经验和强大的程序调试工具。调试程序是查找、发现和纠正错误的有效途径。

1. 错误类型

- 语法错误: 在编译过程中由计算机发现的不符合语言词法或语法规则的错误,如: 输入语句“READ *,i,j”写成“READ *,i+j”,将产生一个语法错误;

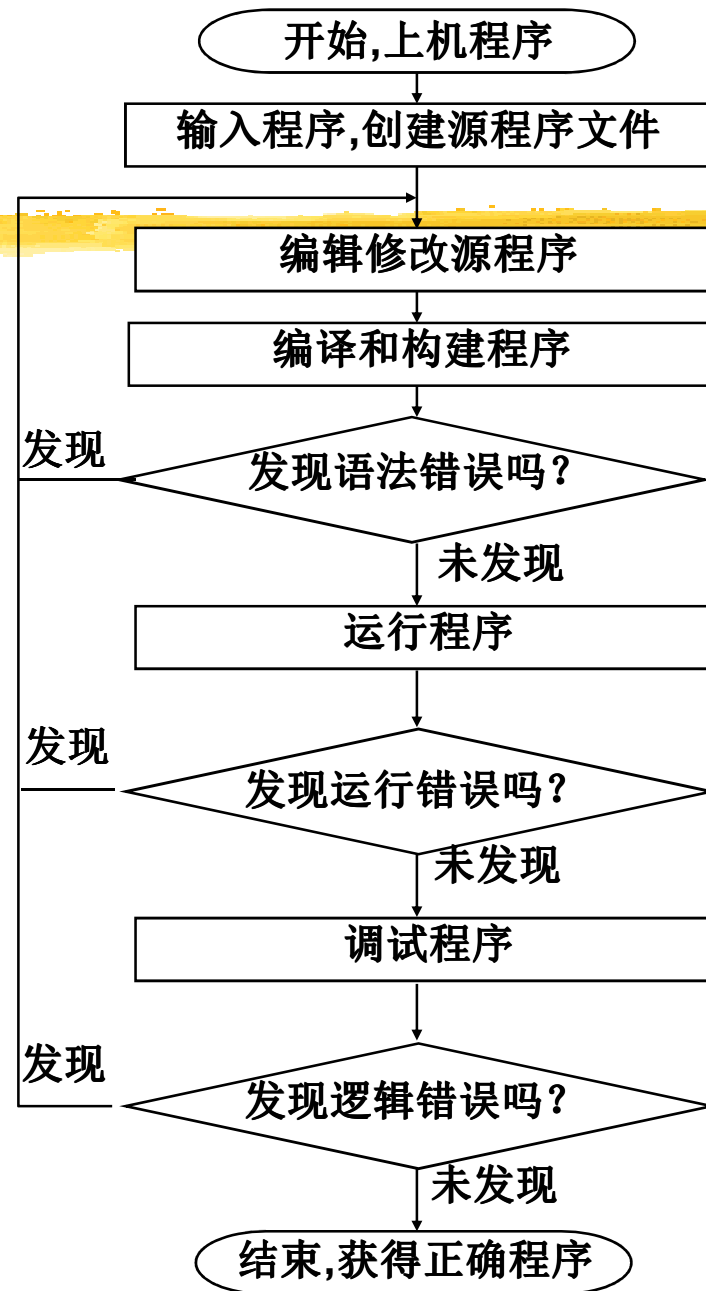
- 运行错误: 在运行过程中由计算机发现的不符合语言语义规则的错误,如: 表达式*i/j*在运行中*j*为0,将产生一个运行错误;

- 逻辑错误: 计算机不能发现的错误,如: 表达式*i+j*错误写成*i-j*,将产生一个逻辑错误。

2. 查错方法

3. 调试工具


调试程序基本步骤如图2-43所示。



程序调试步骤说明

2.6 Developer Studio解题过程

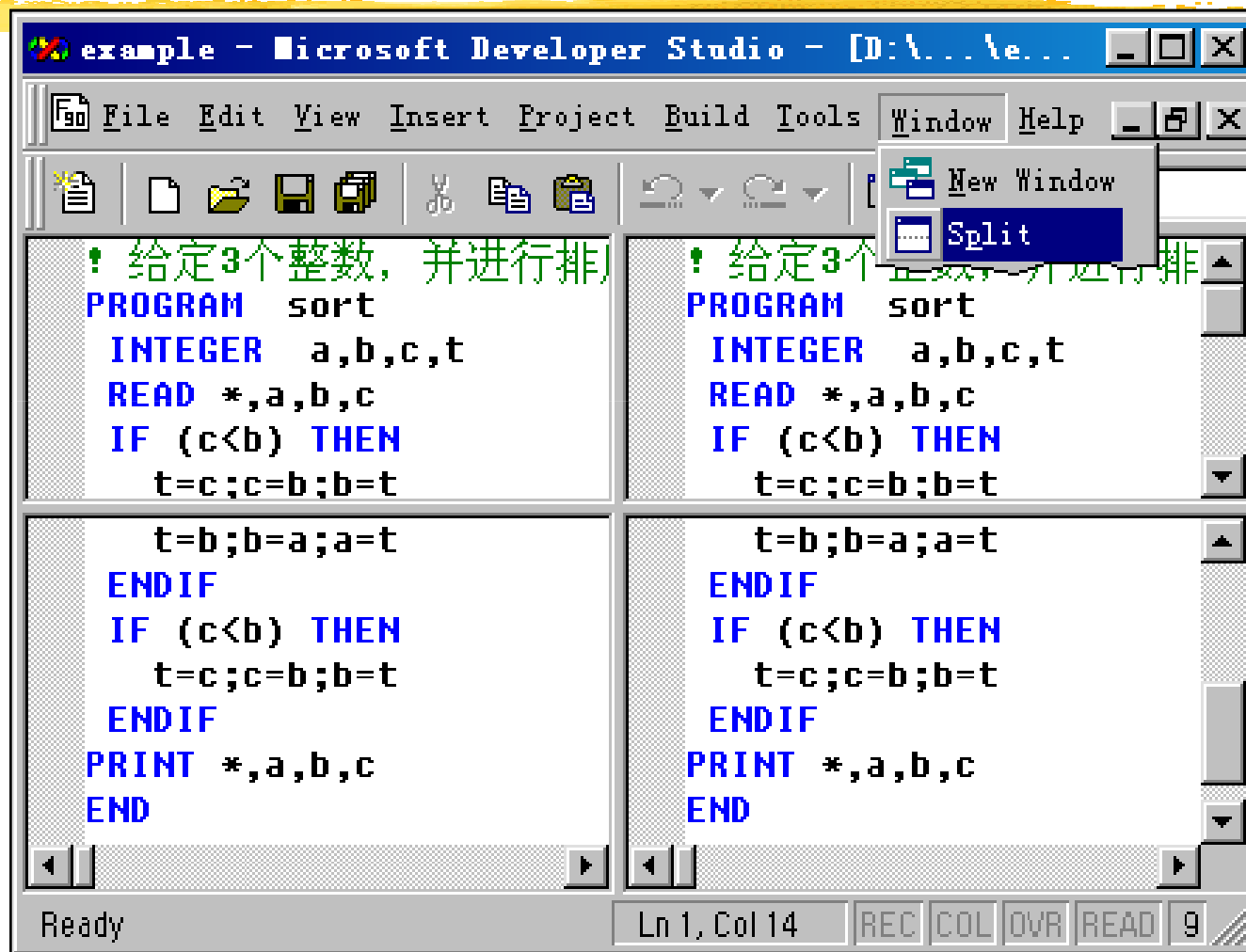


- ◆ 概述
- ◆ 前期准备
- ◆ 创建工作空间
- ◆ 创建项目
- ◆ 创建源程序文件
- ◆ 创建辅助文档
- ◆ 编译源程序
- ◆ 构建可执行程序
- ◆ 运行可执行程序
- ◆ 调试程序
- ◆ 程序多区域显示 

当编写的程序比较大时,在有限的屏幕区域内浏览和查找程序中有关部位显得很不方便,给调试程序带来一定困难。为了解决这一问题,开发环境提供了程序多区域显示功能,通过该功能,用户可同时浏览和查询程序的不同部位。

执行菜单“**Windows/Split**”,可将窗口分裂成四个区域,可同时查询程序四个部位。一、二区域或三、四区域可同时左右移动程序文本,一、三区域或二、四区域可同时上下移动程序文本。如图2-46所示。

程序多区域显示说明



2.7 MS-DOS命令行开发环境简介



- ◆ 概述
- ◆ MS-DOS环境启动
- ◆ 创建源程序文件
- ◆ 编译源程序文件
- ◆ 连接OBJ文件
- ◆ 运行可执行文件

➡ Digital Visual FORTRAN 5.0不但提供了功能强大的可视化集成开发环境Microsoft Developer Studio,而且也提供了基于文本的MS-DOS命令行开发环境。在MS-DOS环境中也可编辑、编译、构建、调试和运行程序,许多在Microsoft Developer Studio中能做的事,在MS-DOS环境中也能做。

在MS-DOS环境中开发FORTRAN程序也有一定优点,其优点是界面简洁、操作方便,对于熟悉DOS环境的程序设计人员可在MS-DOS环境中开发FORTRAN 90程序。

安装Digital Visual FORTRAN 5.0时,在C盘(或其它系统安装盘)创建一个文件夹: **PFS4**,该文件夹中复制一些有关MS-DOS环境中开发程序需要的系统文件,如,编译程序: **Fl32.exe**、连接程序: **Link.exe**、一些管理程序和若干**Lib库文件**。



2.7 MS-DOS命令行开发环境简介



- ◆ 概述
- ◆ MS-DOS环境启动
- ◆ 创建源程序文件
- ◆ 编译源程序文件
- ◆ 连接OBJ文件
- ◆ 运行可执行文件

Windows系统提供一个文本编辑器(程序): EDIT, 使用EDIT文本编辑器可创建、编辑、输入、修改源程序。

创建、编辑、输入、修改源程序步骤:

- 执行MD和CD命令创建或进入相关目录。
- 执行EDIT程序,打开文本编辑器窗口,编辑和输入源程序文本,如图2-47所示。
- 保存源程序文件,打开“FILE”下拉菜单,选择保存菜单项,指定文件名,如: AA4.F90。



2.7 MS-DOS命令行开发环境简介



- ◆ 概述
- ◆ MS-DOS环境启动
- ◆ 创建源程序文件
- ◆ 编译源程序文件 →
- ◆ 连接OBJ文件
- ◆ 运行可执行文件

FL32.exe为FORTRAN编译程序,运行该程序可生成**obj**文件和**exe**文件,如键入“**FL32 AA4.F90**”,可生成**AA4.OBJ**和**AA4.EXE**文件。如有错误,将在窗口内显示错误信息,用户可根据错误信息纠正错误,修改程序,继续编译和运行,直到无错误为止。执行**DF.exe**程序直接编译生成可执行程序文件(exe文件)。



2.7 MS-DOS命令行开发环境简介



- ◆ 概述
- ◆ MS-DOS环境启动
- ◆ 创建源程序文件
- ◆ 编译源程序文件
- ◆ 连接OBJ文件
- ◆ 运行可执行文件

Link.exe为FORTRAN语言的连接装配程序,运行该程序可将obj文件连接装配成exe文件,如键入“Link AA4.obj”,可生成AA4.exe文件。



2.7 MS-DOS命令行开发环境简介



- ◆ 概述
- ◆ MS-DOS环境启动
- ◆ 创建源程序文件
- ◆ 编译源程序文件
- ◆ 连接OBJ文件
- ◆ 运行可执行文件

运行可执行程序文件,输入数据,获得运行结果,如键入“AA4”,即可运行AA4程序。





习题二

FORTRAN90程序

示例

```
!Calculate Cylinder Volume
PROGRAM Cylinder
REAL V,r,h
READ*,r,h
 $V=3.14*r*r*h$ 
PRINT *,'Volume=',V
END PROGRAM Cylinder
```

```
!Calculate Taper Volume
PROGRAM Taper
REAL V,r,h
READ*,r,h
 $V=3.14*r*r*h/3$ 
PRINT *,'Volume=',V
END PROGRAM Taper
```



h e

e n d

文件名格式：班级 学号 姓名 简略实验名称

邮件标题同文件名

Any questions please 发送至

xingzhengwu@163.com