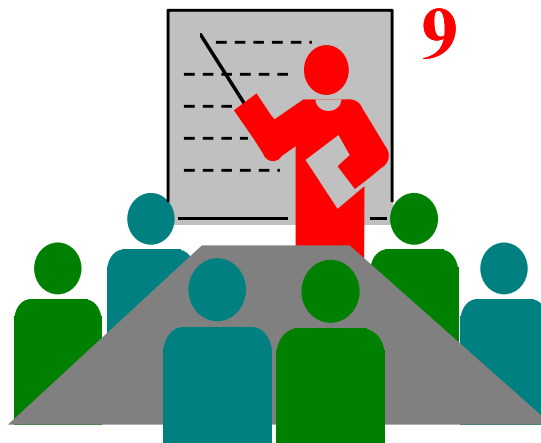


第三章 FORTRAN90基础知识

1. 字符集
2. 名称
3. 关键字
4. 程序单元
5. 语句
6. 书写格式
7. 数据类型
8. 兼容性
9. 习题三



3.1 字符集



- ◆ 概述
- ◆ 字符集
- ◆ 说明



FORTRAN 90源程序是一个符合FORTRAN 90词法、语法和语义规定的特殊文本,它由FORTRAN 90字符集规定的字符按照词法、语法和语义规定进行描述。



3.1 字符集



- ◆ 概述
- ◆ 字符集
- ◆ 说明

FORTRAN 90字符集由下面四部分字符组成:

1. 英文字母

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 。

a b c d e f g h i j k l m n o p q r s t u v w x y z 。

2. 阿拉伯数字: 0 1 2 3 4 5 6 7 8 9。

3. 下划线: “_”。

4. 特殊字符

空格 <Tab> + - * / () , . ' : ! " % & ; < > ? \$ 。



3.1 字符集



- ◆ 概述
- ◆ 字符集
- ◆ 说明

说明:

① 除字符型常量和Hollerith常量外,源程序中词法和语法描述不区分字母的大小写,如语句“INTEGER I”和语句“integer i”是等价的。

② FORTRAN 90字符集以外的可打印字符,只能出现在注释、字符常量、Hollerith常量、字符串编辑符和输入输出记录中。

③ 字符: _ ! " % & ; > < ?为FORTRAN 90新增字符。



3.2 名称



- ◆ 概述
- ◆ 语法描述
- ◆ 示例



名称在FORTRAN 77中称“**符号名**”,在其它语言中称“**标识符**”。名称在程序中用来标识有关实体(如:变量、命名常量、函数、过程、程序单元、公用块等)。

```
PROGRAM main
  INTEGER length,width,area,price
  READ*,length,width
  area=mul(length,width)
  price=mul(4000,area)
  PRINT*,'总地价: ',price,'元'
END
FUNCTION mul(x,y)
  mul=x*y
END FUNCTION
```



3.2 名称



◆ 概述

◆ 语法描述

◆ 示例

$\langle \text{名称} \rangle \rightarrow \langle \text{英文字母} \rangle \{ \langle \text{英文字母} \rangle | \langle \text{数字} \rangle | \text{"_"} | \text{"\$"} \}$

说明:

- ① 名称只能由英文字母、数字、下划线符“_”和美元符号“\$”组成。
- ② 名称第一个字符必须是英文字母。
- ③ 名称不能超过31个字符。



3.2 名称



- ◆ 概述
- ◆ 语法描述
- ◆ 示例



[例3.1]判定下列名称,哪些是合法名称? 哪些是非法名称?
解释非法名称错误原因。

Number, Max, X-Y-Z, PrOgRaM, FIND_IT, read, x, 8q, a.5, _wrong, a3b7, U.S.A., DR.WANG, china, total, x\$y\$z, \$abcd, r a t e。

解: **合法名称有**: Number, Max, PrOgRaM, FIND_IT, read, x, a3b7, china, total, x\$y\$z。

非法名称有: X-Y-Z, 8q, a.5, _wrong, U.S.A., DR.WANG, \$abcd, r a t e。

表3-1 给出非法名称错误原因。

下面是几个使用合法名称的语句例子。

INTEGER total !total命名了一个整型变量

SUBROUTINE example !example命名了一个过程

PROGRAM area !area命名了一个程序单元

Lable: DO I=1,N !Lable命名了一个DO循环



名称示例表3-1

非法名称	错误原因	非法名称	错误原因
X=Y=Z	名称中含减号“-”	U.S.A.	名称中含小数点“.”
8q	“8”作为第一字符	DR.WANG	名称中含小数点“.”
a.5	名称中含小数点“.”	\$abcd	“\$”作为第一字符
wrong	“”作为第一字符	r a t e	名称中含空格

3.3 关键字



- ◆ 概述
- ◆ 说明
- ◆ 示例



关键字是FORTRAN 90中用于描述语句语法成分的特
定字符串。



3.3 关键字



- ◆ 概述
- ◆ 说明
- ◆ 示例

- 关键字都有特定的含义,在描述中有具体的位置要求,不能改变或移位,否则将产生语法错误。如语句“PROGRAM main”中的关键字“PROGRAM”的含义是说明源程序是一个主程序单元,位于语句首部,如果写成“PROGRAMMING main”或“main PROGRAM”将产生语法错误。
- FORTRAN 90对关键字不予保留,即允许其关键字作为其它实体的名称(变量名、数组名、函数名、程序名等),可将PROGRAM、INTEGER、REAL、READ、PRINT、WRITE、DO、END、SUBROUTINE、FUNCTION等作为变量名、数组名、函数名或程序名。编译程序根据上下文来识别一个字符串是关键字,还是实体名称。如语句“PROGRAM PROGRAM”中第一个“PROGRAM”将被编译程序识别为关键字,第二个“PROGRAM”将被识别为实体名称(主程序单元名)。
- 一个应避免使用关键字作为实体名称,因为使用关键字作为实体名称会导致程序难以理解和阅读,降低程序的可读性。



3.3 关键字



- ◆ 概述
- ◆ 说明
- ◆ 示例



```
PROGRAM main
  INTEGER length,width,area,price
  READ*,length,width
  area=mul(length,width)
  price=mul(4000,area)
  PRINT*,'总地价: ',price,'元'
END
FUNCTION mul(x,y)
  mul=x*y
END FUNCTION
```



3.4 语句



◆ 概述

◆ 排列次序

→ FORTRAN 90语句分为两类:

- **可执行语句**: 表示程序要完成某个操作。
- **非执行语句**: 描述程序的某种属性。

```
PROGRAM main
  INTEGER length,width,area,price
  READ*,length,width
  area=length*width
  price=4000*area
  PRINT*,'总地价: ',price,'元'
END
```



3.4 语句



- ◆ 概述
- ◆ 排列次序

程序中语句要求按规定次序排列,即语句位置不能随意放置,否则将产生语法错误,如INTEGER语句和PRINT语句的先后顺序就不能颠倒。语句排列次序规定如表3-2所示,表中给出了全部语句的排列次序要求。语句的排列次序与执行次序不完全相同。

```
PROGRAM main
  INTEGER length,width,area,price
  READ*,length,width
  area=length*width
  price=4000*area
  PRINT*,'总地价: ',price,'元'
END
```



3.5 程序单元



- ◆ 概述 → 一个F90程序可由多个程序单元组成(至少一个主程序单元)。
- ◆ 分类 程序单元由**数据声明**和相关操作(语句)构成,必须用**END**语句
- ◆ 描述 结束。
- ◆ 示例

```
PROGRAM main
  INTEGER length,width,area,price
  READ*,length,width
  area=length*width
  price=4000*area
  PRINT*,'总地价: ',price,'元'
END
```



3.5 程序单元



- ◆ 概述
- ◆ 分类
- ◆ 描述
- ◆ 示例

FORTRAN 90规定有四种程序单元：**主程序单元、外部子程序单元、模块单元和块数据单元**。每个程序单元可分别编译，一起构建(连编)产生可执行程序。

- **外部子程序单元**：不包含在主程序单元、模块单元和其它外部子程序单元中的函数或例行子程序所构成的程序单元，其中外部函数和外部例行子程序能被FORTRAN 90程序中其它程序单元调用执行。由FUNCTION或SUBROUTINE语句开始。
 - **模块单元**：能被其它程序单元访问的一组定义(数据类型定义、过程定义)、过程接口定义)所构成的程序单元，其中模块子程序允许本模块或其它程序单元调用执行。由MODULE语句开始。
 - **块数据单元**：为命名公用块中变量指定初始值的程序单元。由BLOCK DATA语句开始。
 - **主程序单元**：程序中被首次执行的程序单元，程序运行的入口点，其它程序单元执行的启动器。由PROGRAM语句开始，可缺省。
- 主程序单元、外部子程序单元和模块单元可以包含内部子程序，包含内部子程序的程序单元称为宿主(Host)，内部子程序能被其宿主或宿主内的其它子程序调用执行。



3.5 程序单元



<程序> → <主程序单元> { <外部子程序单元> | <模块单元> | <块数据单元> }

- ◆ 概述
- ◆ 分类
- ◆ 描述
- ◆ 示例



<主程序单元> → [<PROGRAM语句>]
[<说明部分>]
[<操作部分>]
[<内部子程序部分>]
<END语句>

```
PROGRAM main
  INTEGER length,width,area,price
  READ*,length,width
  area=length*width
  price=4000*area
  PRINT*, '总地价: ',price,'元'
END
```



3.5 程序单元



- ◆ 概述
- ◆ 分类
- ◆ 描述
- ◆ 示例

[例3.2]编写一个包含下列语句的主程序单元,给出所有可能的主程序单元结构。

```
INTEGER a,b,c
```

```
a=10;b=20
```

```
c=a+b
```

```
PRINT *, ' a+b=',c
```

解: 有5种程序单元结构, 如图3-1所示。



程序单元程序示例

```
INTEGER a,b,c  
a=10;b=20  
c=a+b  
PRINT *,'a+b=',c  
END
```

```
INTEGER a,b,c  
a=10;b=20  
c=a+b  
PRINT *,'a+b=',c  
END PROGRAM
```

```
PROGRAM add  
INTEGER a,b,c  
a=10;b=20  
c=a+b  
PRINT *,'a+b=',c  
END
```

```
PROGRAM add  
INTEGER a,b,c  
a=10;b=20  
c=a+b  
PRINT *,'a+b=',c  
END PROGRAM
```

```
PROGRAM add  
INTEGER a,b,c  
a=10;b=20  
c=a+b  
PRINT *,'a+b=',c  
END PROGRAM add
```

3.6 书写格式



- ◆ 概述 → ● 每种程序设计语言对书写格式都有特定要求。
- ◆ 标志符 ● 传统FORTRAN语言,采用固定格式。
- ◆ 语句标号 ● FORTRAN 90采用固定格式和自由格式。推荐自由格式。
- ◆ 自由格式
- ◆ 固定格式



3.6 书写格式



- ◆ 概述
- ◆ 标志符
- ◆ 语句标号
- ◆ 自由格式
- ◆ 固定格式

书写格式都定义若干标志符,指出文本的意义。
F90RTRAN90定义3个通用标志符:

- 注释标志符 “!”

可出现在程序单元任何位置,表示其后至行末为注释文本。注释行只起解释和说明作用,对程序单元的编译没有影响。一行均为空格也是一个注释行。

- 语句分隔标志符 “;”

可出现在两条语句之间,表示左右为两条独立的语句。允许一行书写多条语句,用符号 “;” 分隔语句。

- 续行标志符 “&”

可出现在一条语句内部,表示前后文本属于一条语句。一条长语句可分多行书写,中间需用 “&” 连接,出现在前一续行的末尾。在Digital Visual FORTRAN 5.0中,续行可达511行。

表3-4给出标志符及其含义。(示例)



3.6 书写格式



- ◆ 概述
- ◆ 标志符
- ◆ 语句标号
- ◆ 自由格式
- ◆ 固定格式

- 语句前面可设置**语句标号**(即语句编号)。
- **语句标号**作用是可改善和提高程序可读性;便于其他语句访问该语句,获取信息或转移控制。语句标号可放在任何完整语句前面,但不属于后面的语句。
- **语句标号**是长度不超过5位十进制正整数,不允许全0。空语句不能加语句标号。语句标号的声明具有唯一性,在同一程序单元中不能声明相同的语句标号。

```
1 PROGRAM main
2  INTEGER length,width,area,price
3  READ*,length,width
4  area=length*width
5  price=4000*area
6  PRINT*,'总地价: ',price,'元'
7  END
```



3.6 书写格式



- ◆ 概述
- ◆ 标志符
- ◆ 语句标号
- ◆ 自由格式
- ◆ 固定格式

- 注释标志符“!”，在FORTRAN 90中,注释行必须以感叹号“!”开始可出现在程序单元任何位置，表示其后至行末为注释文本。
- 语句分隔标志符“;”，FORTRAN允许一行写多条语句，所有书写格式都定义了语句分隔符“;”用于分隔语句。
- 续行标志符“&”，FORTRAN 90中，“&”作为一个续行的开始，一条长语句可分多行书写，中间需用续行标志符连接。在自由格式中，续行标志只能出现在前一续行的末尾。
- 语句标号：语句标号可出现在一条语句的前面，长度不超过5位十进制数，不允许全为0。
- 在自由格式中，语句在一行中的位置不受限制。语句长度可达132个字符。



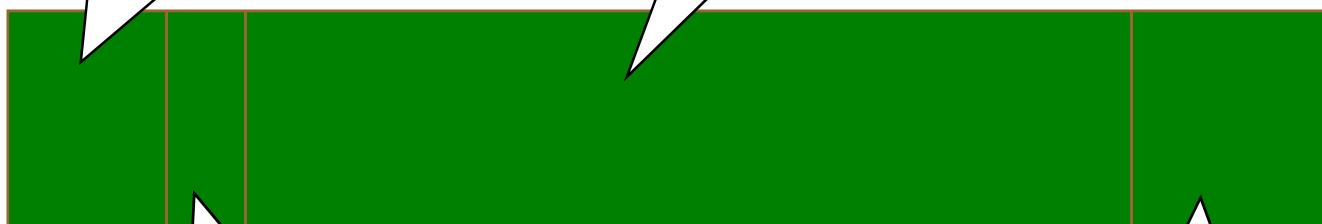
3.6 书写格式



- ◆ 概述
- ◆ 标志符
- ◆ 语句标号
- ◆ 自由格式
- ◆ 固定格式

标号区：第1~5列，用来书写标号，标号为5位以内的整数，数字间有无空格完全等效。

在7~72列中任何列开始写，用来写FORTRAN语句。



续行标志区：续行必须在第6列标志出来。当写入一个非零非空字符时，表示此行是上一行的续行。

在73~80列可以用来写对语句的注释。



3.7 数据类型



- ◆ 概述
- ◆ 数据
- ◆ 数据类型

计算机系统的处理对象是数据(信息),离开了数据,计算机就失去了应有的价值。**数据、数据表示及数据类型**在计算机科学应用(特别是程序设计语言和程序设计)中占有重要位置。

```
PROGRAM main
  INTEGER length,width,area,price
  READ*,length,width
  area=length*width
  price=4000*area
  PRINT*,'总地价: ',price,'元'
END
```



3.7 数据类型



- ◆ 概述
- ◆ 数据
- ◆ 数据类型



数据是自然界中客观事物的符号表示。在计算机科学中,一切能够被计算机接受和处理的对象(信息)均看成是数据。如:面积、周长、体积、距离、速度等都是计算机处理的数据。随着计算机技术的发展,数据的内涵也在不断扩展。数据由早期的整数、实数、复数等简单数据,发展到现在字符串、文本、图形、图像、声音等复杂数据。

数据分**数值数据**和**非数值数据**。

- 数值数据是能直接进行算术运算(科学计算)的数据。如:整数、实数、复数等。
- 非数值计算是不能直接进行算术运算,一般进行查询、检索、分类和处理等操作的数据。如:字符串、文本、表格、图形、图像、声音等。



3.7 数据类型



- ◆ 概述
- ◆ 数据
- ◆ 数据表示
- ◆ 数据类型

不同类型数据有不同的特性,其处理方式不同,取值范围不同。在处理数据前,必须说明数据的特性。数据类型是说明数据特性的常用形式,在语言中被广泛采用。

数据类型是数据取值范围和在其上定义的一组操作的总称。数据类型规定了一种数据取值范围和一组操作。

数据类型有四个性质:

- 每个数据类型有一个唯一的名称。内部数据类型名称由F语言预先定义。派生类型名称根据类型定义语句定义。
- 每个数据类型规定了一个取值范围(值的集合)。如: 8位整型数据类型的取值范围是-128至127。
- 每个数据类型规定了其常量数据的表示方法,如: 12.5。
- 每个数据类型规定了一组操作。如整型规定的操作有: 加+、减-、乘*和除/操作。

FORTRAN 90具有丰富的数据类型,有内部数据类型、自定义数据类型、数组和指针类型等。利用丰富的数据类型可设计和编写出功能强大的FORTRAN 90程序。

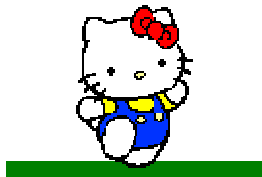
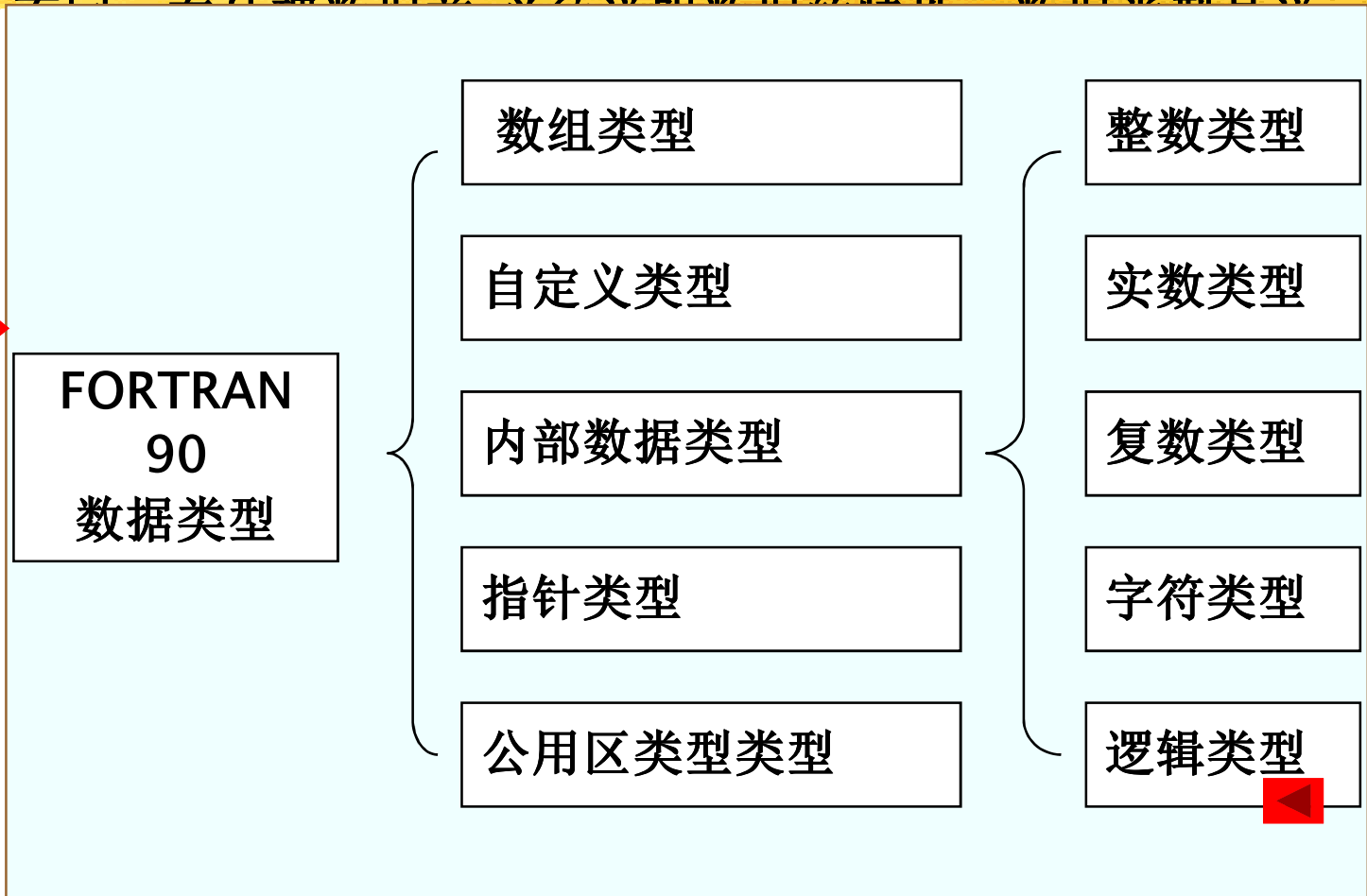


3.7 数据类型



不同类型数据有不同的特性,其处理方式不同,取值范围不同

- ◆ 概述
- ◆ 数据
- ◆ 数据表示
- ◆ 数据类型



3.9 习题三



3.9 习题三



3.9 习题三





The

end

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