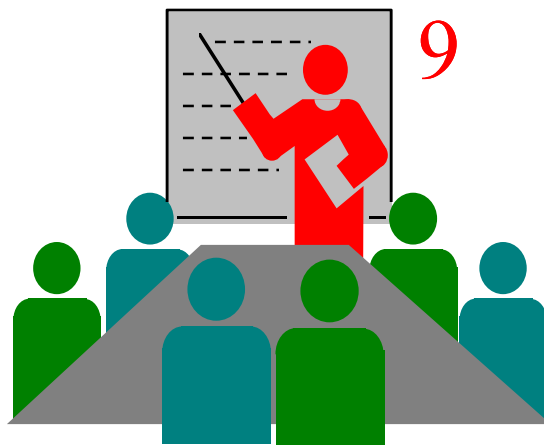


第九章 子程序

1. 概述
2. 标准子程序
3. 语句函数
4. 内部子程序
5. 子程序形参与实参之间的数据传递关系
6. 重入口语句(ENTRY语句)
7. 递归子程序
8. 外部子程序
9. 习题九



9.1 概述



- ◆ 概述
- ◆ 情况1
- ◆ 情况2
- ◆ 情况3
- ◆ 说明



- 使用函数和子例行程序能编写结构清晰、层次分明、短小简洁的好程序，是编写高质量、高水平、高效率程序的有效手段。
- 通常在设计和编写程序时，会遇到以下三种情况：

(1) 同一程序内多次出现相同或相似的运算或处理过程。

(2) 不同程序内多次出现相同或相似的运算或处理过程。

(3) 程序复杂。



9.1 概述



◆ 概述

◆ 情况1

◆ 情况2

◆ 情况3

◆ 说明

在设计和编写程序时,程序中不同部位会多次出现相同或相似的运算或处理过程(程序代码)。

例如: 求五边形ABCDE的面积问题,如图9-1所示。

已知: 五边形5条边长为: a 、 b 、 c 、 d 、 e , 两条对角线为: f 、 g 。五边形面积计算问题变为3个三角形面积计算问题, 3个三角形面积 $area1$ 、 $area2$ 、 $area3$ 之和就是五边形面积 $area$ 。 三角形面积 A 求解公式:

$$p = (x + y + z) / 2$$
$$A = \sqrt{p(p - x)(p - y)(p - z)}$$

● 一般方法求解(程序)。

● 函数方法求解(程序)。

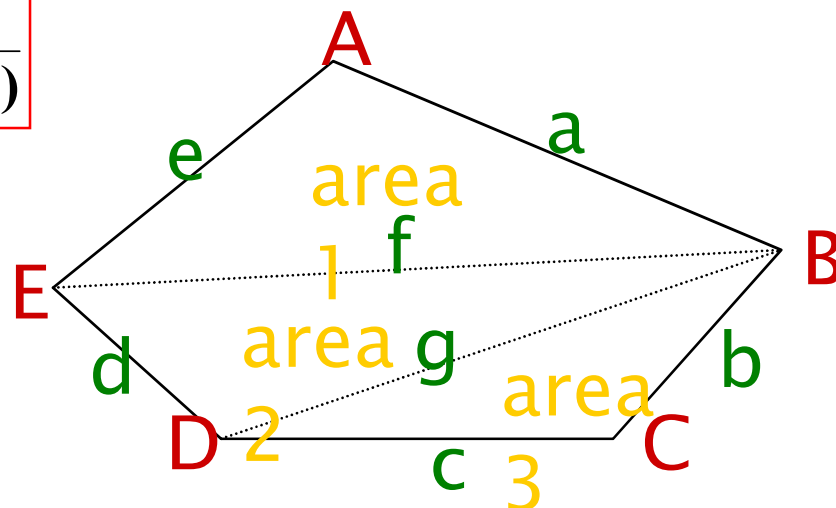


图 9-1 五边形



9.1 概述



◆ 概述

◆ 情况1

◆ 情况2

◆ 情况3

◆ 说明

在设计和编写程序时,不同程序中不同部位会多次出现相同或相似的运算或处理过程(程序代码)。

● 问题一: 某房地产公司买下一块呈三角形形状的地盘准备盖住宅小区,地价为每平方米1000元,求这块地的总价。

求解问题的核心是计算三角形面积。用一般方法求解(程序)。

● 问题二: 某单位有一个呈三棱台形状的水箱,平均每天消耗2水箱水,水价为每立方米8角钱,计算每天需支付的水费。

求解问题的核心是计算三角形面积。用一般方法求解(程序)。

● 用函数方法求解两问题(程序)。



9.1 概述



- ◆ 概述
- ◆ 情况1
- ◆ 情况2
- ◆ 情况3
- ◆ 说明



对于比较复杂的问题,模块化和结构化程序设计方法是常用的程序设计方法,使用这些方法可大大降低问题的复杂性,可使用模块化和结构化程序设计方法设计和编写大型、复杂、高质量程序。

例如: 输入一批测试数据(30个),对其排序,输出排序结果数据。

- 一般方法求解(程序)。
- 结构化程序设计方法及函数方法求解(分层结构图、程序)。



9.1 概述



- ◆ 概述
 - ◆ 情况1
 - ◆ 情况2
 - ◆ 情况3
 - ◆ 说明
- FORTRAN 90子程序，也称过程，划分为两类：**函数**和**子例行程序**。
 - 函数是指产生一个运算结果且通过操作数调用执行的子程序。
 - 子例行程序是指可产生多个处理结果且通过**CALL**语句调用执行的子程序。
 - 每类子程序又划分为**标准子程序**和**用户自定义子程序**。
 - 用户自定义子程序又划分为**内部子程序**、**外部子程序**和**模块子程序**。
 - FORTRAN 90的子程序分类如图9-3所示。



← Back

9.1 概述

◆ 概述

◆ 情况1

◆ 情况2

◆ 情况3

◆ 说明

在设计和编写程序时,程序中不同部位会多次出现相同或相似的运算或处理过程(程序代码)。

例如: 求五边形ABCDE的面积问题,如图9-1所示。

已知: 五边形5条边长为: a 、 b 、 c 、 d 、 e ,两条对角线为: f 、 g 。五边形面积计算问题变为3个三角形面积计算问题,3个三角形面积 $area1$ 、 $area2$ 、 $area3$ 之和就是五边形面积 $area$ 。

三角形面积A求解公式: 三边长为 x,y,z 。

- 一般方法求解(程序)。
- 函数方法求解(程序)。

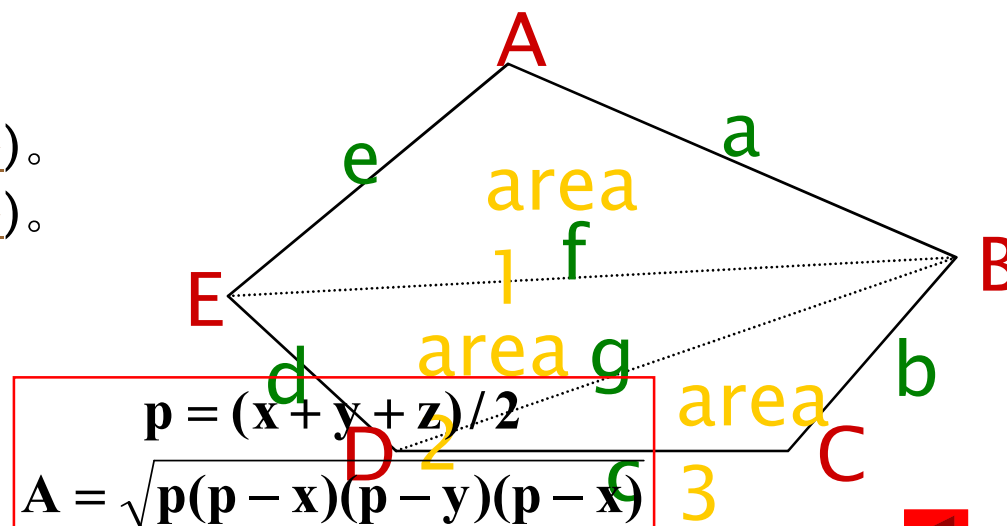


图9-1 五边形



```

PROGRAM pentagon
REAL a,b,c,d,e,f,g
REAL area1,area2,area3,area,p,q
!输入五边形5个边长和两条对角线
READ *,a,b,c,d,e,f,g
!计算三角形ABE的面积area1
p=(a+f+e)/2
q=p*(p-a)*(p-f)*(p-e)
area1=SQRT(q)
!计算三角形BDE的面积area2
p=(g+d+f)/2
q=p*(p-g)*(p-d)*(p-f)
area2=SQRT(q)
!计算三角形BCD的面积area3
p=(b+c+g)/2
q=p*(p-b)*(p-c)*(p-g)
area3=SQRT(q)
area=area1+area2+area3
PRINT *,‘五边形面积:',area
END

```


9.1 概述



◆ 概述

◆ 情况1

◆ 情况2

◆ 情况3

◆ 说明

在设计和编写程序时,程序中不同部位会多次出现相同或相似的运算或处理过程(程序代码)。

例如: 求五边形ABCDE的面积问题,如图9-1所示。

已知: 五边形5条边长为: a 、 b 、 c 、 d 、 e ,两条对角线为: f 、 g 。五边形面积计算问题变为3个三角形面积计算问题,3个三角形面积 $area1$ 、 $area2$ 、 $area3$ 之和就是五边形面积 $area$ 。

三角形面积A求解公式: 三边长为 x, y, z 。

$$p = (x + y + z) / 2$$

$$A = \sqrt{p(p-x)(p-y)(p-z)}$$

● 一般方法求解(程序)。

● 函数方法求解(程序)。

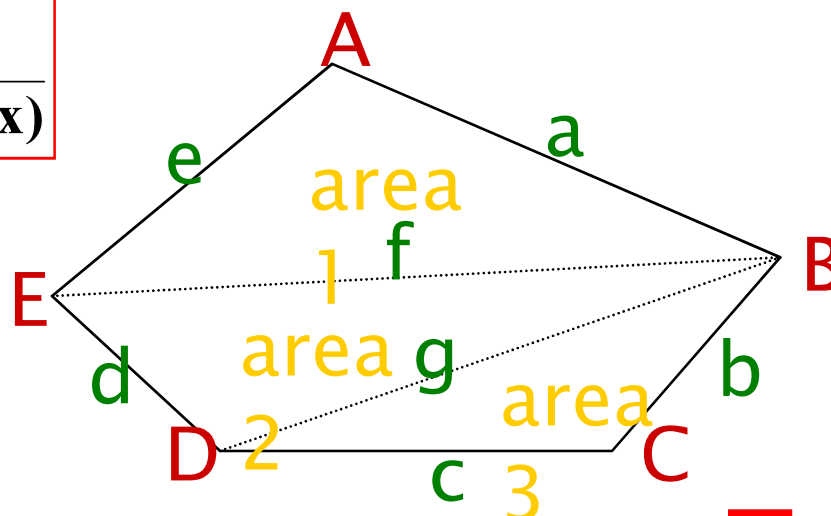


图9-1 五边形



```
PROGRAM pentagon
```

```
REAL a,b,c,d,e,f,g
```

```
REAL area
```

```
!输入五边形5条边长和两条对角线
```

```
READ *,a,b,c,d,e,f,g
```

```
!计算五边形面积area
```

```
area=t_area(a,f,e)+t_area(g,d,f)+t_area(b,c,g)
```

```
PRINT *,'五边形面积为: ',area
```

```
CONTAINS
```

```
!定义计算任意三角形面积函数t-area
```

```
FUNCTION t_area(a,b,c)
```

```
  REAL t_area,a,b,c,p,q
```

```
  p=(a+b+c)/2
```

```
  q=p*(p-a)*(p-b)*(p-c)
```

```
  t_area=SQRT(q)
```

```
END FUNCTION t_area
```

```
END
```

} 计算任意三角形

9.1 概述



◆ 概述

◆ 情况1

◆ 情况2

◆ 情况3

◆ 说明

在设计和编写程序时,不同程序中不同部位会多次出现相同或相似的运算或处理过程(程序代码)。

● **问题一**: 某房地产公司买下一块呈三角形形状的地盘准备盖住宅小区,地价为每平方米1000元,求这块地的总价。

求解问题的核心是计算三角形面积。用一般方法求解(程序)。

● **问题二**: 某单位有一个呈三棱台形状的水箱,平均每天消耗2水箱水,水价为每立方米8角钱,计算每天需支付的水费。

求解问题的核心是计算三角形面积。用一般方法求解(程序)。

● 用函数方法求解两问题(程序)。





```
PROGRAM total_prices
REAL a,b,c,area,p,q,t_prices
!输入三角形边长a,b,c
READ *,a,b,c
!计算三角形面积area
p=(a+b+c)/2
q=p*(p-a)*(p-b)*(p-c)
area=SQRT(q)
!计算总价t_prices
t_prices=area*1000
PRINT *,'地块总价为: ',t_prices
END
```

9.1 概述



◆ 概述

◆ 情况1

◆ 情况2

◆ 情况3

◆ 说明

在设计和编写程序时,不同程序中不同部位会多次出现相同或相似的运算或处理过程(程序代码)。

● **问题一**: 某房地产公司买下一块呈三角形形状的地盘准备盖住宅小区,地价为每平方米1000元,求这块地的总价。

求解问题的核心是计算三角形面积。用一般方法求解(程序)。

● **问题二**: 某单位有一个呈三棱台形状的水箱,平均每天消耗2水箱水,水价为每立方米8角钱,计算每天需支付的水费。

求解问题的核心是计算三角形面积。用一般方法求解(程序)。

● 用函数方法求解两问题(程序)。





```
PROGRAM water_prices
REAL a,b,c,h,area,cubage,p,q,w_prices
!输入三棱台底面三角形边长a,b,c和三棱台高h
READ *,a,b,c,h
!计算三角形面积area
p=(a+b+c)/2
q=p*(p-a)*(p-b)*(p-c)
area=SQRT(q)
!计算三棱台容积cubage
cubage=area*h
!计算水价w_prices
PRINT *,'每天需支付水价为: ',t_prices
END
```

9.1 概述



◆ 概述

◆ 情况1

◆ 情况2

◆ 情况3

◆ 说明

在设计和编写程序时,不同程序中不同部位会多次出现相同或相似的运算或处理过程(程序代码)。

● **问题一**: 某房地产公司买下一块呈三角形形状的地盘准备盖住宅小区,地价为每平方米1000元,求这块地的总价。

求解问题的核心是计算三角形面积。用一般方法求解(程序)。

● **问题二**: 某单位有一个呈三棱台形状的水箱,平均每天消耗2水箱水,水价为每立方米8角钱,计算每天需支付的水费。

求解问题的核心是计算三角形面积。用一般方法求解(程序)。

● 用函数方法求解两问题(程序)。

!定义外部函数area

FUNCTION area(x,y,z)

REAL area,x,y,z,p,q

!计算三角形面积area

$p = (x + y + z) / 2$

$q = p * (p - x) * (p - y) * (p - z)$

area = SQRT(q)

END FUNCTION area



!使用函数求解问题一程序，**area**为外部函数

PROGRAM total_prices

EXTERNAL area !声明**area**为外部函数

REAL a,b,c,t_prices

READ *,a,b,c !输入三角形边长**a,b,c**

t_prices=area(a,b,c)*1000 !计算总价**t_prices**

PRINT *,'地块总价为: ',t_prices

END

!使用函数求解问题二程序，**area**为外部函数

PROGRAM water_prices

EXTERNAL area !声明**area**为外部函数

REAL a,b,c,h,cubage,w_prices

READ *,a,b,c,h !输入棱台底面三角形边长**a,b,c**和台高**h**

cubage=area(a,b,c)*h !计算三棱台容积**cubage**

w_prices=cubage*0.8*2 !计算水价**w_prices**

PRINT *,'每天需支付水价为: ',w_prices

END

9.1 概述



- ◆ 概述
- ◆ 情况1
- ◆ 情况2
- ◆ 情况3
- ◆ 说明



对于比较复杂的问题,设计和编写程序需要采用科学的程序设计方法。模块化和结构化程序设计方法是常用的程序设计方法,使用这些方法可大大降低问题的复杂性,使复杂问题的求解变得相对容易。使用函数和子例行程序可使用模块化和结构化程序设计方法设计和编写大型、复杂、高质量程序。

例如: 输入一批测试数据(30个),对其排序,输出排序结果数据。

- 一般方法求解(程序)。
- 结构化程序设计方法及函数方法求解(分层结构图、程序)。

对复杂问题逐层进行任务分解,得到问题分层结构图,如图9-2所示。根据分层结构图,采用自顶向下或自底向上方法逐层设计和编写程序。最顶层问题用主程序实现,其余层子问题用函数或子例行程序实现,最后将主程序和若干函数或子例行程序组装成完整的可运行程序。

设计和编写主程序时,可假定下一层子问题已经通过函数或子例行程序实现,暂不考虑其具体实现内容,假定: “输入”子问题由子例行程序 **input(A)** 实现,功能是从键盘输入数据至数组 **A**; “排序”子问题由子例行程序 **sort(A)** 实现,功能是对数组 **A** 进行排序,排序结果在 **A** 中; “输出”子问题由子例行程序 **output(A)** 实现,功能是输出数组 **A** 的内容。



```
PROGRAM sort
  PARAMETER (N=30)
  REAL dat(N)
  DO I=1,N      !输入N个数据
    READ *,dat(I)
  END
  DO I=1,N-1    !对N个原始数据进行排序
    DO J=I+1,N
      IF (dat(J)>dat(I)) THEN
        t=dat(J);dat(J)=dat(I);dat(I)=t
      ENDIF
    ENDDO
  ENDDO
  DO I=1,N      !输出排序后的N个数据
    PRINT *,dat(I)
  END
END
END
```

9.1 概述



- ◆ 概述
- ◆ 情况1
- ◆ 情况2
- ◆ 情况3
- ◆ 说明



对于比较复杂的问题,设计和编写程序需要采用科学的程序设计方法。模块化和结构化程序设计方法是常用的程序设计方法,使用这些方法可大大降低问题的复杂性,使复杂问题的求解变得相对容易。使用函数和子例行程序可使用模块化和结构化程序设计方法设计和编写大型、复杂、高质量程序。

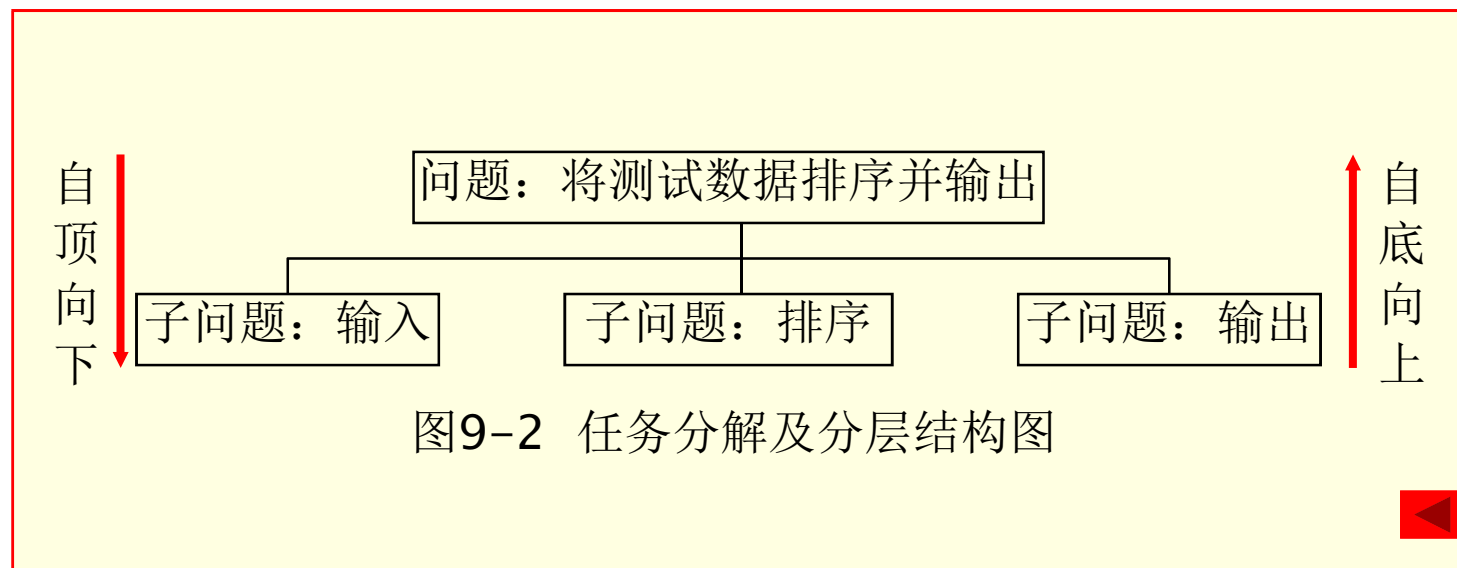
例如: 输入一批测试数据(30个),对其排序,输出排序结果数据。

- 一般方法求解(程序)。
- 结构化程序设计方法及函数方法求解(分层结构图、程序)。

对复杂问题逐层进行任务分解,得到问题分层结构图,如图9-2所示。根据分层结构图,采用自顶向下或自底向上方法逐层设计和编写程序。最顶层问题用主程序实现,其余层子问题用函数或子例行程序实现,最后将主程序和若干函数或子例行程序组装成完整的可运行程序。

设计和编写主程序时,可假定下一层子问题已经通过函数或子例行程序实现,暂不考虑其具体实现内容,假定: “输入”子问题由子例行程序 **input(A)** 实现,功能是从键盘输入数据至数组A; “排序”子问题由子例行程序 **sort(A)** 实现,功能是对数组A进行排序,排序结果在A中; “输出”子问题由子例行程序 **output(A)** 实现,功能是输出数组A的内容。





9.1 概述



- ◆ 概述
- ◆ 情况1
- ◆ 情况2
- ◆ 情况3
- ◆ 说明



对于比较复杂的问题,设计和编写程序需要采用科学的程序设计方法。模块化和结构化程序设计方法是常用的程序设计方法,使用这些方法可大大降低问题的复杂性,使复杂问题的求解变得相对容易。使用函数和子例行程序可使用模块化和结构化程序设计方法设计和编写大型、复杂、高质量程序。

例如:输入一批测试数据(30个),对其排序,输出排序结果数据。

- 一般方法求解(程序)。
- 结构化程序设计方法及函数方法求解(分层结构图、程序)。

对复杂问题逐层进行任务分解,得到问题分层结构图,如图9-2所示。根据分层结构图,采用自顶向下或自底向上方法逐层设计和编写程序。最顶层问题用主程序实现,其余层子问题用函数或子例行程序实现,最后将主程序和若干函数或子例行程序组装成完整的可运行程序。

设计和编写主程序时,可假定下一层子问题已经通过函数或子例行程序实现,暂不考虑其具体实现内容,假定:“输入”子问题由子例行程序 **input(A)** 实现,功能是从键盘输入数据至数组 **A**;“排序”子问题由子例行程序 **sort(A)** 实现,功能是对数组 **A** 进行排序,排序结果在 **A** 中;“输出”子问题由子例行程序 **output(A)** 实现,功能是输出数组 **A** 的内容。



!主程序

```
PROGRAM main
  PARAMETER (N=30)
  REAL dat(N)
  CALL input(dat) !输入N个数据
  CALL sort(dat) !对N个原始数据进行排序
  CALL output(dat) !输出排序后的N个数据
CONTAINS
...
END
```

!输入子例行程序

```
SUBROUTINE input(A)
  REAL A(N)
  DO I=1,N
    READ *,A(I)
  ENDDO
END      SUBROUTINE
input
```

!排序子例行程序

```
SUBROUTINE sort(A)
  REAL A(N),t
  DO I=1,N-1
    DO J=I+1,N
      IF (A(J)>A(I)) THEN
        t=A(J);A(J)=A(I);A(I)=t
      ENDIF
    ENDDO
  ENDDO
END SUBROUTINE sort
```

!输出子例行程序

```
SUBROUTINE
output(A)
  REAL A(N)
  DO I=1,N
    PRINT *,A(I)
  ENDDO
END      SUBROUTINE
output
```

9.1 概述



◆ 概述

◆ 情况1

◆ 情况2

◆ 情况3

◆ 说明

● FORTRAN 90子程序，也称过程，划分为两类：函数和子例行程序。

● 函数是指产生一个运算结果且通过操作数调用执行的子程序。

如上面计算三角形面积的子程序area就是一个函数。

● 子例行程序是指可产生多个处理结果且通过CALL语句调用执行的子程序。

如上面完成输入操作的子程序input(A)就是一个子例行程序。

● 每类子程序又划分为标准子程序和用户自定义子程序。

● 用户自定义子程序又划分为内部子程序、外部子程序和模块子程序。

● FORTRAN 90的子程序分类如图9-3所示。



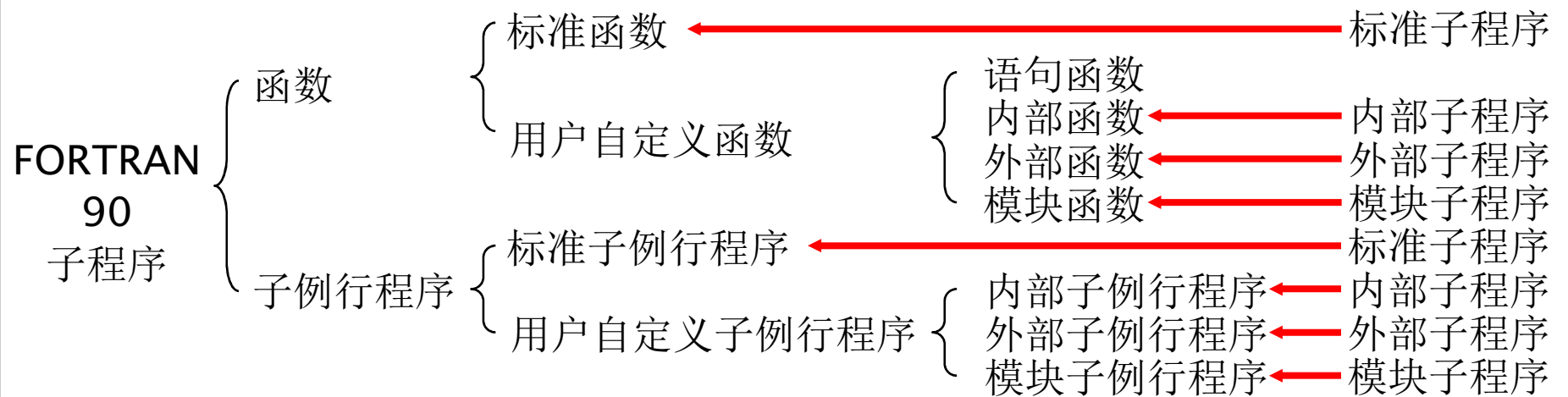


图9-3 FORTRAN 90子程序分类

9.2 标准子程序



- ◆ 概述
- ◆ 标准函数
- ◆ 注意事项1
- ◆ 标准子例行程序
- ◆ 注意事项2



- 标准子程序用于求解是现实生活中比较常见的问题。

如：求绝对值、三角函数值、指数值、获取当前系统日期、获取随机数等。

- 为了提高子程序的执行效率,子程序一般用执行效率比较高的汇编语言编写,并汇编成机器语言目标代码,它们被集中保存在一个子程序库中,供用户直接使用。



9.2 标准子程序



- ◆ 概述
- ◆ 标准函数
- ◆ 注意事项1
- ◆ 标准子例行程序
- ◆ 注意事项2

标准函数有时称系统函数、内部函数,FORTRAN 90的编译系统提供130多个标准函数。有以下几大

类:

- 三角函数计算
- 数值计算
- 数据类型转换
- 数据类型查询
- 随机数生成
- 日期时间处理
- 字符串处理
- 地址计算
- 位运算

标准函数可作为表达式中操作数进行调用,并参与表达式运算。调用标准函数时,只需写出函数名,后面括号中给出符合要求的所要计算的变元值。例如:

已知: $x=3.0, y=4.0$, 计算表达式
 $2.5 * \text{SQRT}(x * x + y * y) + \text{ABS}(-8.5 * 3)$,
结果为38.0。



9.2 标准子程序



- ◆ 概述
- ◆ 标准函数
- ◆ 注意事项1
- ◆ 标准子例行程序
- ◆ 注意事项2



使用标准函数需要注意以下几点:

- 标准函数对变元的个数和类型有明确的规定和要求。

要求有一个变元,如函数 $\text{SIN}(x)$,要求两个变元,如函数 $\text{MOD}(m,n)$, 要求多个变元,如函数 $\text{MAX}(x_1,x_2,\dots,x_n)$ 。

- 标准函数的函数值只有一个,且有明确的类型规定。

多数函数函数值类型与变元类型相同,如函数 $\text{SQRT}(9.0\text{D}0)=3.0\text{D}0$,类型都为双精度。

- 对于“弧度”和“角度”单位,使用不同的三角函数。
- 函数变元必须写在函数名后括号内,变元可是常数、变量或表达式,少数标准函数变元要求变量,如 $\text{LOC}(v)$

9.2 标准子程序



- ◆ 概述
- ◆ 标准函数
- ◆ 注意事项1
- ◆ 标准子例行程序
- ◆ 注意事项2

标准子例行程序也称系统子例行程序,FORTRAN 90的编译系统提供20多个标准子例行程序。

有以下几大类:

- 程序控制(EXIT、SLEEPQQ)
- 文件管理(PACKTIMEQQ、UNPACKTIMEQQ)
- 随机数生成(RANDOM、RANDOM_NUMBER)
- 日期处理(DATE、GETDAT、TIME、GETTIM)
- 数组处理(SUM、SORTQQ、COUNT、SIZE)



调用标准子例行程序时,需通过关键字CALL完成,其后给出被调用标准子例行程序名,并在后面的括号中给出符合要求的变元值(一般为变量、数组、结构体,也可为常数或表达式)。例如:

CALL GETDAT(year,month,day),获取当前系统日期,年、月和日数据,通过变元year、month和day返回。

9.2 标准子程序



- ◆ 概述
- ◆ 标准函数
- ◆ 注意事项1
- ◆ 标准子例行程序
- ◆ 注意事项2

使用标准子例行程序要注意以下几点:

- 标准子例行程序对变元的个数和类型有明确的规定和要求。
- 标准子例行程序可以产生多个结果数据,通过变元返回调用程序,也可不产生返回数据。
- 如果标准子例行程序调用时没有变元,则子程序名字后的一对括号可省略。
如**CALL EXIT()**和**CALL EXIT**是等价的。
- 标准子例行程序的变元必须写在子程序名后面的括号内。



← Back

9.3 语句函数



- ◆ 概述
- ◆ 定义
- ◆ 调用
- ◆ 说明
- ◆ 示例1
- ◆ 示例2
- ◆ 示例3

在求解实际问题时,特别是在求解有关科学与工程数值计算问题时,常常会遇到计算一些数学表达式,如:

$$\begin{array}{ccc} \sqrt{a^2 + b^2 + 2ab\cos\alpha} & \sqrt{x^2 + y^2} & \frac{x^2}{\sqrt{1 + 2x + x^2}} \\ \sqrt{p(p-a)(p-b)(p-c)} & x^2 + 2x + 1 & \frac{1}{2}ab\sin\alpha \end{array}$$

这些数学表达式在程序中多次出现,需要重复书写它们,任何数学表达式实际上都可表示成一元、二元或多元函数关系,即 $F(x_1, x_2, \dots, x_n)$, 这些函数关系如果没有对应的标准函数,则需要用户设法定义新的函数来实现它们。

(示例)



9.3 语句函数



- ◆ 概述
- ◆ 定义
- ◆ 调用
- ◆ 说明
- ◆ 示例1
- ◆ 示例2
- ◆ 示例3



一般形式为: $F(x_1, x_2, \dots, x_n) = \langle \text{表达式} \rangle$

F为函数名, $x_1, x_2, \dots, x_n$ 为虚拟变元(或称形式参数或哑元)。

要求: 不能重名,只能为普通变量名。类型可用隐含规则定义,也可用类型声明语句定义,但须放在该语句函数定义语句之前。



9.3 语句函数



- ◆ 概述
- ◆ 定义
- ◆ 调用
- ◆ 说明
- ◆ 示例1
- ◆ 示例2
- ◆ 示例3



调用一般形式为：

<语句函数名>([<实在参数表>])

说明：

- <语句函数名>同语句函数定义语句中的语句函数名相同。
- <实在参数表>为实在参数或实参,其个数、次序和类型应与定义语句中形式参数一致。实在参数之间用逗号分隔。
- 实参可是常数、变量、数组元素、函数名、表达式。



9.3 语句函数



◆ 概述

◆ 定义

◆ 调用

◆ 说明

◆ 示例1

◆ 示例2

◆ 示例3

使用语句函数的一些规定:

- 一个语句函数只能在一行内定义。如果太长,可续行书写。
- 语句函数只有一个函数返回值。
- 定义语句函数,右端表达式中可调用已定义语句函数或标准函数,但不能调用该语句函数自己,即不能递归调用。

$H(x1,x2)=x1*x1+x2*x2$

$F(x1,x2)=SQRT(H(x1,x2))$

- 语句函数的形式参数不允许为数组或下标变量。

$DIMENSION A(10),B(2,3)$

$F1(x,y)=x+A(1)+y-B(1,2)$!表达式中可以引用数组元素

$F(C)=C+100$

- 语句函数名属于局部变量,只能在定义它的程序单位中引用。
- 语句函数可无形参,称为无参函数,圆括号不能省略,即: $F()$ 。



9.3 语句函数



◆ 概述

◆ 定义

◆ 调用

◆ 说明

◆ 示例1

◆ 示例2

◆ 示例3

[例9.3]计算空间一点P(x,y,z)的方向弦,其计算公式如下:

$$\cos \alpha = \frac{x}{\sqrt{x^2 + y^2 + z^2}} \quad \cos \beta = \frac{y}{\sqrt{x^2 + y^2 + z^2}} \quad \cos \gamma = \frac{z}{\sqrt{x^2 + y^2 + z^2}}$$

其中: α 、 β 、 γ 为点P到原点直线与坐标轴x、y、z的夹角。

```
PROGRAM F931
```

```
F(a,b,c,r)=r / SQRT(a*a+b*b+c*c)
```

```
WRITE(*,*) '输入 x,y,z:'
```

```
READ(*,*) x,y,z
```

```
DA=F(x,y,z,x)
```

```
DB=F(x,y,z,y)
```

```
DG=F(x,y,z,z)
```

```
WRITE(*,*) DA,DB,DG
```

```
END
```



9.3 语句函数



[例9.4]已知 $\alpha=30^\circ$, $\beta=38^\circ$, $\gamma=58^\circ$,编写程序求:

$$y = \frac{1 + \sin \alpha + \sin^2 \alpha}{1 + \cos \alpha} \left(\sqrt{\frac{1 + \sin \beta + \sin^2 \beta}{1 + \cos \beta} - \frac{1 + \sin \gamma + \sin^2 \gamma}{1 + \cos \gamma}} \right)$$

- ◆ 概述
- ◆ 定义
- ◆ 调用
- ◆ 说明
- ◆ 示例1
- ◆ 示例2
- ◆ 示例3

```
PROGRAM F94
```

```
PARAMETER(pi=3.1415926)
```

```
F(A)=1.0+SIN(A*pi/180)+SIN(A*pi/180)**2
```

```
→ F1(A)=F(A)/(1.0+COS(A*pi/180))
```

```
WRITE(*,*) ' 输入 $\alpha,\beta,\gamma$  角度(度数):'
```

```
READ(*,*) alfa,bata,gama
```

```
y=F1(alfa)*SQRT(F1(bata)-F1(gama))
```

```
WRITE(*,10) INT(alfa),INT(bata),INT(gama),y
```

```
10 FORMAT(1x,'y(',I2,',',I2,',',I2,')=',F10.7)
```

```
END
```



9.3 语句函数



- ◆ 概述
- ◆ 定义
- ◆ 调用
- ◆ 说明
- ◆ 示例1
- ◆ 示例2
- ◆ 示例3

[例9.5] 试用下列公式编写求面积的程序。

$$F \approx \frac{B-A}{N} \left\{ \frac{1}{2} [f(A) + f(B)] + \sum_{i=1}^{N-1} f(x_i) \right\}$$

$$f(x) = e^{-x^2}$$

```
PROGRAM F95
```

```
f(x)=EXP(-x*x)
```

```
READ(*,*)A,B,N
```

```
h=(B-A)/N; sum=(f(A)+f(B))/2.0; x=A+h
```

```
DO i=1, N-1
```

```
    sum=sum+f(x); x=x+h
```

```
ENDDO
```

```
sum=sum*h
```

```
WRITE(*,*) A,B,N,sum
```

```
END
```



← Back

9.3 语句函数



- ◆ 概述
- ◆ 定义
- ◆ 调用
- ◆ 说明
- ◆ 示例1
- ◆ 示例2
- ◆ 示例3

[例9.1] 已知角度 $A=36^{\circ}43'27''$, $B=15^{\circ}24'$, $C=8^{\circ}16'54''$, 计算:

$$S = \cos^2 A - 4 \sin B \cdot \tan C$$

角度 $D_n = x_1^{\circ} x_2' x_3''$ 的弧度转换公式: $D_n = \frac{\pi}{180} (x_1 + \frac{x_2}{60} + \frac{x_3}{3600})$

```
PROGRAM F91
```

```
p1=3.14159/180.0
```

```
A=p1*(36.0+43.0/60+27.0/3600.0)
```

```
B=p1*(15.0+24.0/60)
```

```
C=p1*(8.0+16.0/60+54.0/3600)
```

```
S=COS(A)**2-4.0*SIN(B)*TAN(C)
```

```
WRITE (*,10) A,B,C,S
```

```
10 FORMAT('A=',F10.7,'B=',F10.7,'C=',F10.7,/'S=',F10.7)
```

```
END
```



9.4 内部子程序



- ◆ 概述
- ◆ 内部函数定义
- ◆ 内部函数说明
- ◆ 内部函数调用
- ◆ 内部子例行程序定义
- ◆ 内部子例行程序说明
- ◆ 内部子例行程序调用

- FORTRAN90提供标准子程序、语句函数、内部子程序(内部函数、内部子例行程序)等。

- 内部子程序是包含在程序CONTAINS结构中用户数和子例行程序它们的程序

```
PROGRAM main
  PRINT*,add(5,3)
CONTAINS
  FUNCTION
  add(m,n)
    INTEGER m,n,add
    add=m+n
  END      FUNCTION
add
END PROGRAM main
```

```
PROGRAM main
  INTEGER add
  PRINT*,add(5,3)
END PROGRAM main
FUNCTION add(m,n)
  INTEGER m,n,add
  add=m+n
END FUNCTION add
```

```
FUNCTION add(m,n)
  INTEGER m,n,add
  add=m+n
END FUNCTION add
PROGRAM main
  INTEGER add
  INTEGER*,add(5,3)
END PROGRAM main
```

9.4 内部子程序



- ◆ 概述
- ◆ 内部函数定义
- ◆ 内部函数说明
- ◆ 内部函数调用
- ◆ 内部子例行程序定义
- ◆ 内部子例行程序说明
- ◆ 内部子例行程序调用

- 内部函数子程序,简称内部函数,放在CONTAINS结构中。
- 内部函数只产生一个结果,由函数名返回。
- 内部函数定义一般格式:

<内部函数定义> →
[<类型>] FUNCTION <函数名> ([<形式参数表>])
 {<说明语句部分>}
 {<执行语句部分>}
END FUNCTION [<函数名>]

- 例如:

```
REAL FUNCTION add(m,n)
  INTEGER m,n
  add=m+n
END FUNCTION add
```

(示例1) (示例2)

```
FUNCTION add(m,n)
  INTEGER m,n,add
  add=m+n
END      FUNCTION
add
```


9.4 内部子程序



- ◆ 概述
- ◆ 内部函数定义
- ◆ 内部函数说明
- ◆ 内部函数调用
- ◆ 内部子例行程序定义
- ◆ 内部子例行程序说明
- ◆ 内部子例行程序调用

- 关键字FUNCTION前面可指定内部数据类型,如: INTEGER、REAL、COMPLEX、CHARACTER 或 LOGICAL等。如果没指定,则按I-N规则或类型声明语句声明类型。

- <形式参数表>称为哑元表,无形参,则圆括号不能省略。

- 函数体位于FUNCTION语句和END FUNCTION语句之间。

- 说明部分同主程序,说明形参、变量等类型。

- 执行部分同主程序,进行数据处理。

- 例: 计算 $X=2.5$ 时多项式值

$$P_5(x) = 5.0x^5 + 2.5x^4 + 4.4x + 3.5$$

$$P_3(x) = 3.4x^3 + 1.5x^2 + 1.76x + 3$$

调用计算

pnx(5,A,2.5)

pnx(3,A,2.5)

```
FUNCTION pnx(N,A,X)
INTEGER N
REAL    A(1:N+1)    ,
X, pnx
S=A(1)
DO I=2,N+1
    S=S*X+A(I)
ENDDO
pnx=S
END FUNCTION
```

9.4 内部子程序



- ◆ 概述
- ◆ 内部函数定义
- ◆ 内部函数说明
- ◆ 内部函数调用
- ◆ 内部子例行程序定义
- ◆ 内部子例行程序说明
- ◆ 内部子例行程序调用

- 调用格式要求同标准函数、语句函数。作为操作数用。
- 一般格式： $F(a_1, a_2, \dots, a_n)$ F 为函数名, a_i 为实在参数。
- 实在参数可是表达式、普通变量、下标变量、数组名、内部函数引用、外部函数引用以及外部函数名和子程序名等。
- 如果定义内部函数无形式参数,其引用形式为: $F()$ 。
- (示例)

9.4 内部子程序



- ◆ 概述
- ◆ 内部函数定义
- ◆ 内部函数说明
- ◆ 内部函数调用
- ◆ 内部子例行程序定义
- ◆ 内部子例行程序说明
- ◆ 内部子例行程序调用

● 内部子例行程序,简称内部子程序,也称内部过程,放在CONTAINS结构中。

● 内部子例行程序可产生多个结果,由参数返回。

● 内部子例行程序定义一般格式:

<内部子例行程序定义>→

SUBROUTINE <子程序名> ([<形式参数表>])

{<说明语句部分>}

{<执行语句部分>}

END SUBROUTINE [<子程序名>]

例如:

SUBROUTINE swap(x,y)

INTEGER temp,x,y

temp=x; x=y; y=temp

END SUBROUTINE swap

(示例1)

(示例2)

(示例3)

9.4 内部子程序



- ◆ 概述
- ◆ 内部函数定义
- ◆ 内部函数说明
- ◆ 内部函数调用
- ◆ 内部子例行程序定义
- ◆ 内部子例行程序说明
- ◆ 内部子例行程序调用

- 子例行程序不指定数据类型，子程序名不返回值。
- <形式参数表>称为哑元表，无形参，则圆括号可省略。
- 子程序体位于SUBROUTINE和END SUBROUTINE语句之间。
- 说明部分同主程序，说明函数名、形参、变量等类型。
- 执行部分同主程序，进行数值运算或数据处理。
- 例：计算 $X=2.5$ 时多项式值

$$P_5(x) = 5.0x^5 + 2.5x^4 + 4.4x + 3.5$$

$$P_3(x) = 3.4x^3 + 1.5x^2 + 1.76$$

调用计算

CALL pnf(5,A,2.5,pnx)
PRINT*,pnx
CALL pnf(3,A,2.5,pnx)
PRINT*,pnx

```
SUBROUTINE
pnf(N,A,X,pnx)
INTEGER I,N
REAL A(N+1),X,pnx,S
S=A(1)
DO I=2,N+1
  S=S*X+A(I)
ENDDO
pnx=S
END SUBROUTINE
```

9.4 内部子程序



- ◆ 概述
- ◆ 内部函数定义
- ◆ 内部函数说明
- ◆ 内部函数调用
- ◆ 内部子例行程序定义
- ◆ 内部子例行程序说明
- ◆ 内部子例行程序调用

- 通过CALL语句调用。作为语句使用。
 - 一般格式：CALL P ($a_1, a_2, \dots, a_n$) 或CALL P。
 - 形参和实参的有关要求同内部函数。
 - 如果无形式参数,其调用形式为: CALL P。
 - 内部子例行程序执行过程:
 - 若实在参数为表达式,则先对表达式求值。
 - 实在参数与相应的形式参数结合,将实参值或地址传递给对应形参。
 - 经过形实结合,形式参数获得了所需要的值或地址。
 - 执行子程序体。
 - 返回调用处下一条语句执行。
- (示例1) (示例2)



← Back

9.4 内部子程序



- ◆ 概述
- ◆ 内部函数定义
- ◆ 内部函数说明
- ◆ 内部函数调用
- ◆ 内部子例行程序定义
- ◆ 内部子例行程序说明
- ◆ 内部子例行程序调用

- 内部函数子程序,简称内部函数,放在CONTAINS结构中。
- 内部函数只产生一个结果,由函数名返回。
- 内部函数定义一般格式:

← 内部函数定义 →

```
[<类型>] FUNCTION <函数名> ([<形式参数表>])  
    {<说明语句部分>}  
    {<执行语句部分>}  
END FUNCTION [<函数名>]
```

例如:

```
REAL FUNCTION add(m,n)  
    INTEGER m,n  
    add=m+n  
END FUNCTION add
```

(示例1) (示例2)

```
FUNCTION add(m,n)  
    INTEGER m,n,add  
    add=m+n  
END      FUNCTION  
add
```



使用内部函数计算分段函数。

```
PROGRAM main
  READ*,x; PRINT*,f(x)
CONTAINS
  FUNCTION f(y)
    IF (y>0) THEN
      f=x**2+SQRT(SQRT(1+x))
    ELSEIF (y==0) THEN
      f=0
    ELSE
      f=x**3+SQRT(1-x)
    ENDIF
  END FUNCTION f
END PROGRAM main
```

$$f(x) = \begin{cases} x^2 + \sqrt[4]{1+x} & x > 0 \\ 0 & x = 0 \\ x^3 + \sqrt{1-x} & x < 0 \end{cases}$$

9.4 内部子程序



◆ 概述

◆ 内部函数定义

◆ 内部函数说明

◆ 内部函数调用

◆ 内部子例行程序定义

◆ 内部子例行程序说明

◆ 内部子例行程序调用

- 内部函数子程序,简称内部函数,放在CONTAINS结构中。
- 内部函数只产生一个结果,由函数名返回。
- 内部函数定义一般格式:

<内部函数定义>→

```
[<类型>] FUNCTION <函数名> ([<形式参数表>])  
    {<说明语句部分>}  
    {<执行语句部分>}  
END FUNCTION [<函数名>]
```

例如:

```
REAL FUNCTION add(m,n)  
    INTEGER m,n  
    add=m+n  
END FUNCTION add
```

(示例1) (示例2)

```
FUNCTION add(m,n)  
    INTEGER m,n,add  
    add=m+n  
END    FUNCTION  
add
```


求解二元一次方程组。

$$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases} \quad x = \frac{\Delta x}{\Delta} \quad y = \frac{\Delta y}{\Delta}$$

计算： $\Delta x = \frac{\begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}}{\Delta}$ $\Delta y = \frac{\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}}{\Delta}$ $\Delta = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} = a_1b_2 - a_2b_1$

编写内部函数del求解：x，y，参数l=1，求解x，l=2，求解y
设计算法，编写程序。

已知：二元一次方程组。

$$\begin{cases} x + y = 3 \\ 2x + y = 4 \end{cases}$$

输入数据：1, 1, 3
2, 1, 4

输出结果：1.0 2.0
2.0 1.0 4.0

x=1.0000000 y=2.0000000

9.4 内部子程序



- ◆ 概述
- ◆ 内部函数定义
- ◆ 内部函数说明
- ◆ 内部函数调用
- ◆ 内部子例行程序定义
- ◆ 内部子例行程序说明
- ◆ 内部子例行程序调用

- 内部函数子程序,简称内部函数,放在CONTAINS结构中。
- 内部函数只产生一个结果,由函数名返回。
- 内部函数定义一般格式:

← 内部函数定义 →

```
[<类型>] FUNCTION <函数名> ([<形式参数表>])  
    {<说明语句部分>}  
    {<执行语句部分>}  
END FUNCTION [<函数名>]
```

例如:

```
REAL FUNCTION add(m,n)  
    INTEGER m,n  
    add=m+n  
END FUNCTION add  
(示例1) (示例2)
```

```
FUNCTION add(m,n)  
    INTEGER m,n,add  
    add=m+n  
END FUNCTION  
add
```



定义内部函数del(D,l)	
声明形式参数: D(2,3),l	
定义语句函数F(x1,x2,y1,y2)=x1*y2-x2*y1	
计算D1=F(D(1,1),D(2,1),D(1,2),D(2,2))	
l=l+1	
T	F
D2=F(D(1,3),D(2,3),D(1,2),D(2,2))	D2=F(D(1,1),D(2,1),D(1,3),D(2,3))
del=D2/D1	
返回	

声明数组: A(2,3)
输入二元一次方程组的系数和常数项
x=del(A,1),调用内部函数del求x值
y=del(A,2),调用内部函数del求y值
输出方程组系数及常数项
输出x和y

求解二元一次方程组。

$$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases} \quad x = \frac{\boxed{\text{函数框图}}}{\Delta} \quad y = \frac{\Delta y}{\Delta}$$

计算: $\Delta x = \frac{\begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}}{\Delta} = \frac{c_1 b_2 - c_2 b_1}{\Delta}$ $\Delta y = \frac{\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}}{\Delta} = \frac{a_1 c_2 - a_2 c_1}{\Delta}$

$\Delta = a_1 b_2 - a_2 b_1$

编写内部函数del求解: x, y, 参数l=1, 求解x, l=2, 求解y
设计算法, 编写程序。

已知: 二元一次方程组。

$$\begin{cases} x + y = 3 \\ 2x + y = 4 \end{cases}$$

输入数据: 1, 1, 3
 2, 1, 4

输出结果: 1.0 3.0
 2.0 1.0 4.0

x=1.0000000 主程序框图 0

9.4 内部子程序



- ◆ 概述
- ◆ 内部函数定义
- ◆ 内部函数说明
- ◆ 内部函数调用
- ◆ 内部子例行程序定义
- ◆ 内部子例行程序说明
- ◆ 内部子例行程序调用

- 内部函数子程序,简称内部函数,放在CONTAINS结构中。
- 内部函数只产生一个结果,由函数名返回。
- 内部函数定义一般格式:

<内部函数定义>→

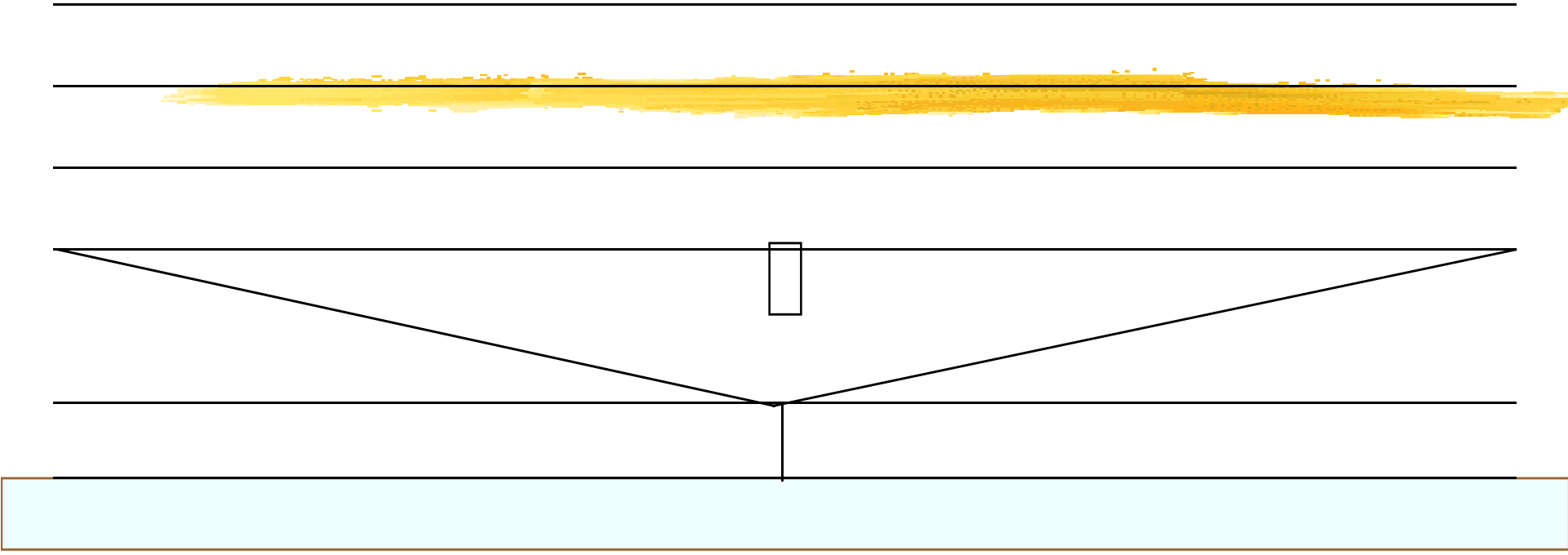
```
[<类型>] FUNCTION <函数名> ([<形式参数表>])  
    {<说明语句部分>}  
    {<执行语句部分>}  
END FUNCTION [<函数名>]
```

例如:

```
REAL FUNCTION add(m,n)  
    INTEGER m,n  
    add=m+n  
END FUNCTION add  
(示例1) (示例2)
```

```
FUNCTION add(m,n)  
    INTEGER m,n,add  
    add=m+n  
END FUNCTION  
add
```





声明数组: $A(2,3)$

输入二元一次方程组的系数和常数项

$x = \text{del}(A, 1)$, 调用内部函数 **del** 求 x 值

$y = \text{del}(A, 2)$, 调用内部函数 **del** 求 y 值

输出方程组系数及常数项

输出 x 和 y

定义内部函数del(D,l)

声明形式参数: D(2,3),l

定义语句函数 $F(x1,x2,y1,y2)=x1*y2-x2*y1$

计算 $D1=F(D(1,1),D(2,1),D(1,2),D(2,2))$

$l=1$

T

F

$D2=F(D(1,3),D(2,3),D(1,2),D(2,2))$ $D2=F(D(1,1),D(2,1),D(1,3),D(2,3))$

$del=D2/D1$

返回

求解二元一次方程组。

$$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases} \quad x = \frac{\Delta x}{\Delta} \quad y = \frac{\Delta y}{\Delta}$$

$$\Delta x = \begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix} \quad \Delta y = \begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix} \quad \Delta = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}$$

计算: $\Delta x = c_1b_2 - c_2b_1$ $\Delta y = a_1c_2 - a_2c_1$ $\Delta = a_1b_2 - a_2b_1$

编写内部函数del求解: x, y, 参数l=1, 求解x, l=2, 求解y

设计算法, 编写程序。

已知: 二元一次方程组。

$$\begin{cases} x + y = 3 \\ 2x + y = 4 \end{cases}$$

输入数据: 1, 1, 3
2, 1, 4

输出结果: 1.0 3.0
2.0 1.0 4.0

x=1.0000000 y=2.0000000


```
PROGRAM F96
DIMENSION A(2,3)
READ(*,*) ((A(I,J),J=1,3),I=1,2)
x=del(A,1); y=del(A,2)  ! 求x,y的值
WRITE(*,100)((A(I,J),J=1,3),I=1,2)
WRITE(*,200) 'x=',x, 'y=',y
100 FORMAT(3X,3F6.0)
CONTAINS
  FUNCTION del(D,I)
    ...
  END FUNCTION del
END
```

定义多项式函数del

```
FUNCTION del(D,I)
  INTEGER I
  REAL D(2,3)
  F(x1,x2,y1,y2)=x1*y2-x2*y1
  D1=F(D(1,1),D(2,1),D(1,2),D(2,2))
  IF (I==1) THEN
    D2=F(D(1,3),D(2,3),D(1,2),D(2,2))
  ELSE
    D2=F(D(1,1),D(2,1),D(1,3),D(2,3))
  ENDIF
  del=D2 / D1
END FUNCTION del
```

9.4 内部子程序



- ◆ 概述
- ◆ 内部函数定义
- ◆ 内部函数说明
- ◆ 内部函数调用
- ◆ 内部子例行程序定义
- ◆ 内部子例行程序说明
- ◆ 内部子例行程序调用

- 调用格式要求同标准函数、语句函数。作为操作数用。
- 一般格式： $F(a_1, a_2, \dots, a_n)$ F 为函数名, a_i 为实在参数。
- 实在参数可是表达式、普通变量、下标变量、数组名、内部函数引用、外部函数引用以及外部函数名和子程序名等。
- 如果定义内部函数无形式参数,其引用形式为: $F()$ 。
- 内部函数执行过程:
 - 若实在参数为表达式,则先对表达式求值。
 - 实在参数与相应的形式参数结合,将实参值传递给对应形参。
 - 经过形实结合,形式参数获得了所需要的值,然后,立即执行内部函数体。
 - 在执行函数体的过程中,若遇到RETURN语句或END FUNCTION语句,则控制返回到引用的地方,并代回所计算的函数值参加表达式的计算。(示例)



```
PROGRAM F971
DIMENSION A(6),B(4)
WRITE(*,*) '请输入A数组的6个元素和B数组的4个元素: '
READ(*,*)(A(I),I=1,6),(B(I),I=1,4)
P5=pnx(5,A,2.5)
P3=pnx(3,B,2.5)
WRITE(*,10) 2.5,P5,2.5,P3
10 FORMAT(1X,'P5(',F3.1,')=',F10.5,'P3(',F3.1,')=',F10.5)
CONTAINS
FUNCTION pnx(N,A,X)
...
END FUNCTION
END
```

输入数据:

5.0,2.5,0.0,0.0,4.4,3.5

3.4,1.5,1.76,3.5

输出结果:

P5(2.5)= 600.43750 P3(2.5)= 70.40000

9.4 内部子程序



- ◆ 概述
- ◆ 内部函数定义
- ◆ 内部函数说明
- ◆ 内部函数调用
- ◆ 内部子例行程序定义
- ◆ 内部子例行程序说明
- ◆ 内部子例行程序调用

● 内部子例行程序,简称内部子程序,也称内部过程,放在CONTAINS结构中。

● 内部子例行程序可产生多个结果,由参数返回。

● 内部子例行程序定义一般格式:

<内部子例行程序定义>→

SUBROUTINE <子程序名> ([<形式参数表>])

{<说明语句部分>}

{<执行语句部分>}

END SUBROUTINE [<子程序名>]

例如:

SUBROUTINE swap(x,y)

INTEGER temp,x,y

temp=x; x=y; y=temp

END SUBROUTINE swap

(示例1)

(示例2)

(示例2)



使用内部子例行程序对A, B, C三数进行排序。

```
PROGRAM sort
  READ*,A,B,C
  IF (A>B) CALL swap(A,B)
  IF (B>C) CALL swap(B,C)
  IF (A>B) CALL swap(A,B)
CONTAINS
  SUBROUTINE swap(x,y)
    REAL temp,x,y
    temp=x; x=y; y=temp
  END SUBROUTINE swap
END PROGRAM sort
```

9.4 内部子程序



- ◆ 概述
- ◆ 内部函数定义
- ◆ 内部函数说明
- ◆ 内部函数调用
- ◆ 内部子例行程序定义
- ◆ 内部子例行程序说明
- ◆ 内部子例行程序调用

● 内部子例行程序,简称内部子程序,也称内部过程,放在CONTAINS结构中。

● 内部子例行程序可产生多个结果,由参数返回。

● 内部子例行程序定义一般格式:

<内部子例行程序定义>→

SUBROUTINE <子程序名> ([<形式参数表>])

{<说明语句部分>}

{<执行语句部分>}

END SUBROUTINE [<子程序名>]

例如:

SUBROUTINE swap(x,y)

INTEGER temp,x,y

temp=x; x=y; y=temp

END SUBROUTINE swap

(示例1)

(示例2)

(示例2)



使用内部子例行程序输出错误信息。

```
PROGRAM main
  READ*,score
  IF (score>=0 .AND. Score<=100) THEN
    IF (score<=60) THEN
      CALL output('该同学成绩不合格。' )
    ELSE
      CALL output('该同学成绩合格。' )
    ENDIF
  ELSE
    CALL output('成绩输入错误。' )
  ENDIF
CONTAINS
  定义子例行程序output
END PROGRAM main
```



```
SUBROUTINE
  output(str)
    CHARACTER*80 str
    PRINT*,str
  END      SUBROUTINE
output
```


9.4 内部子程序



- ◆ 概述
- ◆ 内部函数定义
- ◆ 内部函数说明
- ◆ 内部函数调用
- ◆ 内部子例行程序定义
- ◆ 内部子例行程序说明
- ◆ 内部子例行程序调用



- 内部子例行程序,简称内部子程序,也称内部过程,放在CONTAINS结构中。

- 内部子例行程序可产生多个结果,由参数返回。

- 内部子例行程序定义一般格式:

<内部子例行程序定义>→

SUBROUTINE <子程序名> ([<形式参数表>])

{<说明语句部分>}

{<执行语句部分>}

END SUBROUTINE [<子程序名>]

例如:

SUBROUTINE swap(x,y)

INTEGER temp,x,y

temp=x; x=y; y=temp

END SUBROUTINE swap

(示例1)

(示例2)

(示例2)



使用内部子例行程序求解质因数，结果放数组**P1**中，质因数个数放**N**中。

```
PROGRAM main
  INTEGER M,N,P1(20)
  READ*,M
  CALL prime(M,P1,N)
  PRINT*,M,'的质因数有：'
  PRINT*, P1
CONTAINS
  定义子例行程序prime
END PROGRAM main
```



!求整数质因数的内部子例行程序

```
SUBROUTINE prime(M,P1,N)
  INTEGER P1(20),M,N,M1,I
  N=0; M1=M
  DO I=2,M
    DO WHILE (MOD(M1,I) == 0)
      N=N+1
      P1(N)=I
      M1=M1/I
    ENDDO
  ENDDO
END SUBROUTINE prime
```

9.4 内部子程序

- ◆ 概述
- ◆ 内部函数定义
- ◆ 内部函数说明
- ◆ 内部函数调用
- ◆ 内部子例行程序定义
- ◆ 内部子例行程序说明
- ◆ 内部子例行程序调用

- 通过CALL语句调用。作为语句使用。
 - 一般格式：CALL P ($a_1, a_2, \dots, a_n$) 或CALL P。
 - 形参和实参的有关要求同内部函数。
 - 如果无形式参数,其调用形式为：CALL P。
 - 内部子例行程序执行过程：
 - 若实在参数为表达式,则先对表达式求值。
 - 实在参数与相应的形式参数结合,将实参值或地址传递给对应形参。
 - 经过形实结合,形式参数获得了所需要的值或地址。
 - 执行子程序体。
 - 返回调用处下一条语句执行。
- (示例1) (示例2)

使用内部子例行程序求矩阵**A**的转置矩阵**B**。

```
PROGRAM main
  PARAMETER(N=3)
  INTEGER A(N,N),B(N,N)
  READ*,(A(I,J),J=1,N),I=1,N)
  CALL trans(N,A,B)
  DO I=1,N
    PRINT '(<N>(I3,1X))',A(I,1:N)
  ENDDO
CONTAINS
  定义子例行程序prime
END PROGRAM main
```

!求转置矩阵

```
SUBROUTINE trans(N,A,B)
  INTEGEE A(N,N),B(N,N),N
  INTEGER I,J
  DO I=1,N
    DO J=1,N
      B(J,I)=A(I,J)
    ENDDO
  ENDDO
END SUBROUTINE trans
```

!求转置矩阵

```
SUBROUTINE trans(N,A)
  INTEGEE A(N,N),N,T,I,J
  DO I=1,N
    DO J=I+1,N
      T=A(I,J)
      A(I,J)=A(J,I)
      A(J,I)=T
    ENDDO
  ENDDO
END SUBROUTINE trans
```

9.4 内部子程序



- ◆ 概述
- ◆ 内部函数定义
- ◆ 内部函数说明
- ◆ 内部函数调用
- ◆ 内部子例行程序定义
- ◆ 内部子例行程序说明
- ◆ 内部子例行程序调用

- 通过CALL语句调用。作为语句使用。
 - 一般格式: **CALL P** ($a_1, a_2, \dots, a_n$) 或 **CALL P**。
 - 形参和实参的有关要求同内部函数。
 - 若无形式参数,其调用形式为: **CALL P**。
 - 内部子例行程序执行过程:
 - 若实在参数为表达式,则先对表达式求值。
 - 实在参数与相应的形式参数结合,将实参值或地址传递给对应形参。
 - 经过形实结合,形式参数获得了所需要的值或地址。
 - 执行子程序体。
 - 返回调用处下一条语句执行。
- (示例1) (示例2)



使用内部子例行程序计算多项式。 $P_7(x)=7.8x^7+2.1x^6-5.5x^5+0.23x^3+1.25$

$x=0.1, 0.2, 0.3, \dots, 1.0$

```
PROGRAM F911
  REAL A(8),X(10),PNX(10)
  WRITE(*,*) '输入M(<=10),N(<=7):'
  READ(*,*) M,N
  WRITE(*,*) '输入N+1个系数:'
  READ(*,*) A(1:N+1)
  WRITE(*,*) '输入M个X值:'
  READ (*,*) X(1:M)
  CALL pnf(N,A,M,X,PNX)
  WRITE(*,100) (X(I),PNX(I),I=1,M)
100 FORMAT (1X,'POLY(',F3.1,')=',F14.7,\)
CONTAINS
  定义子例行程序pnf
END PROGRAM main
```

!求多项式

SUBROUTINE pnf(N,A,M,X,PNX)

INTEGER N,M,I,J

REAL A(N+1),X(M),PNX(M)

DO I=1,M

PNX(I)=A(1)

DO J=2,N+1

PNX(I)=PNX(I)*X(I)+A(J)

ENDDO

ENDDO

END SUBROUTINE



输入数据:

输入M(≤ 10),N($N \leq 7$):

10,7

输入N+1个系数:

7.8 2.1 -5.5 0.0 0.23 0.0 0.0 1.0

输入M个X值:

0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

输出结果:

POLY(0.1)= 1.2501779

POLY(0.2)= 1.2503142

POLY(0.3)= 1.2460817

POLY(0.4)= 1.2297812

POLY(0.5)= 1.2006249

.....



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