TI-nspire 系列计算器编程 2——提高：转移与选择

主讲人：王中伟（湖南交通职业技术学院）

【课程目的】

1．理解结构化程序的三种基本结构；

2．掌握无条件转移语句（GOTO 语句）的使用和注意事项；

3．掌握选择结构语句（IF 语句）的使用；

4．理解关系表达式和逻辑表达式。

一、 程序的三种基本结构

过去，由于计算机（或计算器）运行速度和内存的限制，通常把效率放在第一位，为了节约一点运行时间与内存空间，用一些“小技巧”、“小聪明”，使人琢磨不透，程序涩晦难懂。而现在，程序的质量标准是“清晰第一、效率第二”，程序要清晰可读，容易修改，由此提出了“结构化程序设计方法”。

结构化程序设计方法规定了三种基本结构：顺序结构、选择结构、循环结构。

1．顺序结构

见图 1，这是最简单的一种结构，各块是按它们出现的先后顺序执行的。

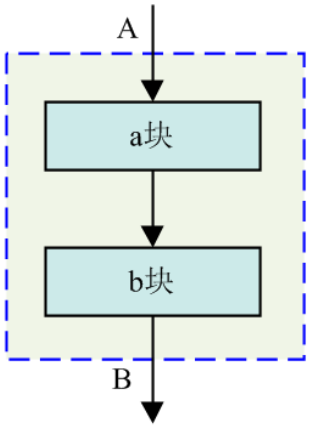

图 1 顺序结构

在前面的“TI-nspire 系列计算器编程 1——入门：从公式到程序”那一讲中，编写了路线圆曲线要素及主点桩号计算程序 ys，其程序结构，就是一个典型的顺序结构（见图 2）。

```
Define ys()=
Prgm
Request "半径",r,0
Request "转角",a,0
Request "交点桩号",jd,0
t:=r·tan(a/2)
l:=r·a·π/180
e:=r·(sec(a/2)-1)
d:=2·t-l
zy:=jd-t
yz:=zy+l
qz:=yz-l/2
Disp "切线长T=",t,",曲线长L=",l
Disp "外距E=",e,",校正值D=",d
Disp "ZY=",zy,",QZ=",qz,",YZ=",yz
EndPrgm
```

原始数据输入模块

↓

曲线要素计算模块

↓

主点桩号计算模块

↓

计算结果输出模块

图 2 路线圆曲线要素及主点桩号计算程序的结构

2．选择结构

见图 3，根据给定条件是否满足决定执行 a 块或 b 块。

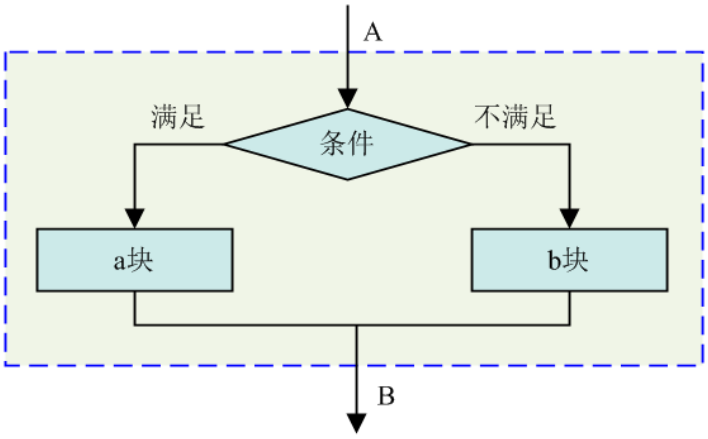

图 3 选择结构

3．循环结构

图 4（a）所示是“当型循环”，即“当给定条件满足时反复执行 a 块，直到条件不满足为止”。图 4（b）所示为“直到型循环”，即“反复执行 a 块，直到给定满足为止”。以上二者的区别是：“当型循环”是先判断后执行，而“直到型循环”是先执行后判断，二者可以

相互转换的。

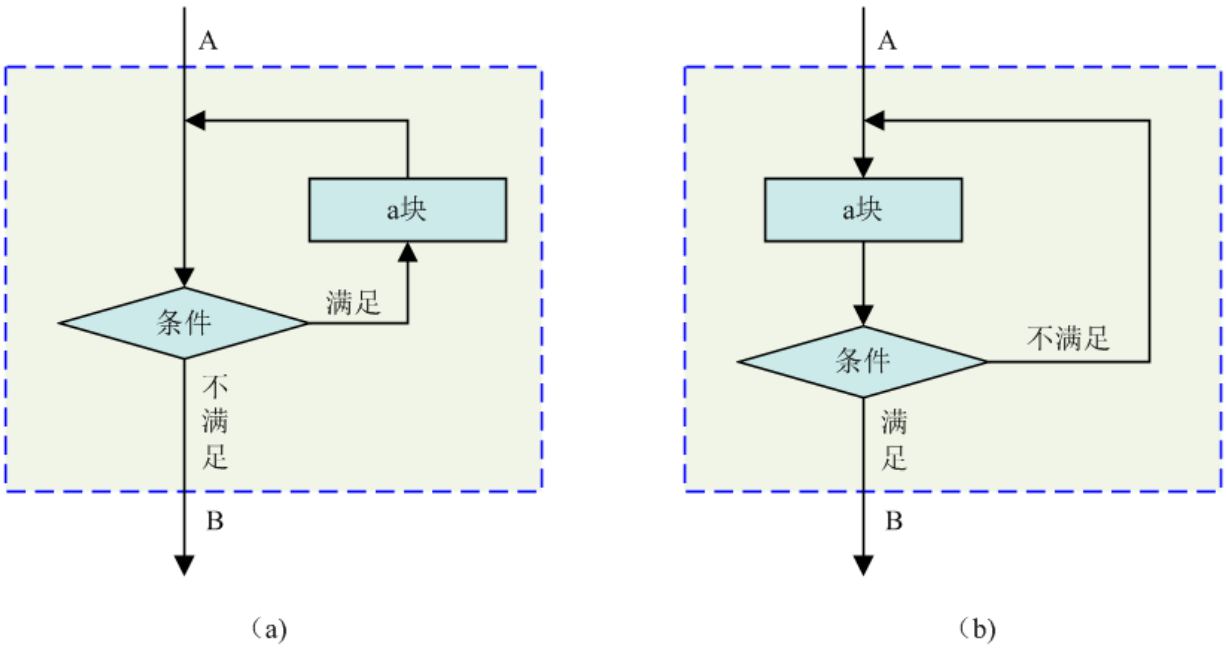

图 4 循环结构

二、 无条件转移语句（Goto~Lbl）

Goto 语句的作用是使流程无条件地转移到指定标记的程序位置去，它使用的一般形式为：

```
Goto 标记
……
Lbl 标记
```

在 TI-nspire 计算器中，标记的定义是和变量的定义规则是一致的，即：

（1）单个字母，如 a，b，c；

（2）多个字母的组合，如 bac，sea；

（3）字母和数字的组合，但第一位必须是字母，如 d34，e5d；

（4）不能使用程序关键字，如 if，stop。

按照以上原则，以下标记不合法：3，5d，6bc，then。

同一个程序中的标记和变量，可以同名。

我们尝试在路线圆曲线要素及主点桩号计算程序 ys 中的相应位置增加这样一对 Goto~Lbl 语句，并执行程序，观察程序运行的流程是否与理解的一致？

```
Define ys()=
Prgm
Request "半径",r,0
Goto a2
Request "转角",a,0
Request "交点桩号",jd,0
t:=r·tan(a/2)
l:=r·a·π/180
e:=r·(sec(a/2)-1)
d:=2·t-l
zy:=jd-t
yz:=zy+l
qz:=yz-l/2
Disp "切线长T=",t,",曲线长L=",l
Lbl a2
Disp "外距E=",e,",校正值D=",d
Disp "ZY=",zy,",QZ=",qz,",YZ=",yz
EndPrgm
```

无条件转移

显然，Goto 语句改变了程序顺序执行语句的性质。结构化程序设计方法反对滥用 Goto 语句，因为它可能导致程序的无规律化、非结构化，一般只允许为了形成选择结构或循环结构而使用 Goto 语句。

三、 选择结构语句（IF 语句）的基本格式

选择结构语句是用 IF 语句，根据条件是否成立（即“真”或者“假”）来实现的。IF 语句常用的基本格式主要有三种。

1．If 语句

格式：If（条件）：语句 1：语句 2……

当满足条件时，执行语句 1、语句 2、……

当不满足条件时，跳过紧挨条件后的语句 1，而执行语句 2 、……

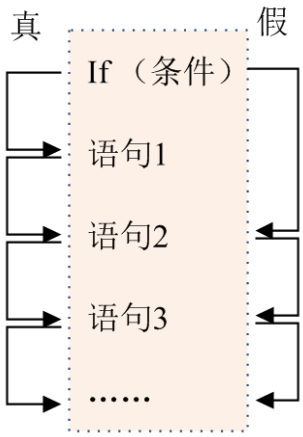

图 5　If 语句的运行流程

【试验 1】尝试对 ys 程序进行修改，赋值给一个变量 n，如果 n 是正数，则不输入半径值。

```
Define ys()=
Prgm
Request "输入数值",n,0
If n≤0
Request "半径",r,0
Request "转角",a,0
Request "交点桩号",jd,0
t:=r· tan(a/2)
```

2．If-Then-EndIf 语句

格式：If （条件） Then: （语句组）: EndIf

当满足条件时，执行关键词“Then”和“EndIf”之间的语句组。

当不满足条件时，则跳过该语句组，而执行关键词“EndIf”后面的语句。

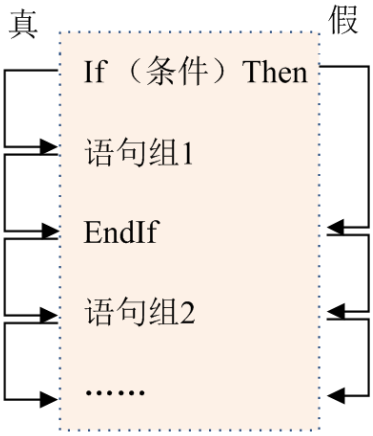

图 6　If-Then-EndIf 语句的运行流程

【试验 2】尝试对 ys 程序进行修改，赋值给一个变量 n，如果 n 是正数，则不输入半径、转角值。

```
Define ys()=
Prgm
Request "输入数值",n,0
If n≤0 Then
Request "半径",r,0
Request "转角",a,0
EndIf
Request "交点桩号",jd,0
t:=r·tan(a/2)
```

3．If-Then-Else-EndIf

格式：If （条件） Then: （语句组 1）: Else: （语句组 2）: EndIf

当满足条件时，执行关键词“Then”和“Else”之间的语句组 1。

当不满足条件时，则执行关键词“Else”和“EndIf”之间的语句组 2。

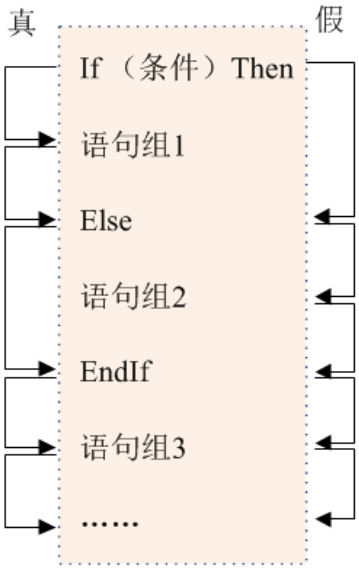

图 7　if-then-else-endif 语句的运行流程

【试验 3】尝试对 ys 程序进行修改，赋值给一个变量 n，如果 n 非正，则输入半径、转角值，否则输入交点桩号。

```
Define ys()=
Prgm
Request "输入数值",n,0
If n≤0 Then
Request "半径",r,0
Request "转角",a,0
Else
Request "交点桩号",jd,0
EndIf
t:=r·tan(a/2)
```

以上三种格式中，括号中的“语句组”可以是一个语句，也可以是多个语句。

选择结构还有更加复杂的多重条件选择（图 8）、IF 语句的嵌套（图 9）等用法，这里不一一举例，读者可以参考有关书籍，自行编写代码尝试。

```
If （条件1）Then
语句组1
ElseIf （条件2）Then
语句组2
ElseIf （条件3）Then
语句组3
……
Else
语句组4
EndIf
```

图 8 多重条件选择

```
If （条件1）Then
语句组1
If （条件2）Then
语句组2
EndIf
语句组3
Else
语句组4
EndIf
```

图 9 If 语句的嵌套

我们现在回头小结一下选择结构语句（IF 语句）的特点：

（1）IF 语句最基本的结构是：**If （条件） Then: （语句组）: EndIf**；

（2）如果满足条件后只有一条语句需要补充执行，则可简化为：**If （条件）**；

（3）在 If~Then~EndIf 语句基本结构的基础上，可根据需要，补充 Else，或者 ElseIf 语句；

（4）在 IF 语句的内部，还可嵌套另外的 IF 语句，并且可以嵌套若干层，使用嵌套关系时，一定要十分仔细，防止出现逻辑错误。

【编程案例 1】编写一个坐标反算的程序，分别输入起点和端点的坐标，计算并显示直线的水平距离和方位角。

验证数据：

起点坐标（m）		端点坐标（m）		距离（m）	方位角
X	Y	X	Y		
1536.861	837.542	2029.555	372.731	677.345	316° 40′ 4.8″
3443.132	2313.237	3494.205	2904.124	593.090	85° 03′ 35.8″
234.454	866.323	146.970	934.543	110.939	142° 03′ 10.5″

程序代码如下：

```
Define axz1()=
Prgm
Request "起点X",x1,0
Request "起点Y",y1,0
Request "端点X",x2,0
Request "端点Y",y2,0
R▸Pr(x2-x1,y2-y1)→dist
R▸Pθ(x2-x1,y2-y1)→fwj
If fwj<0:fwj+360→fwj
Disp "距离=",dist
Disp "方位角=",fwj▸DMS
EndPrgm
```

【编程案例 2】编写一个程序，计算 y：

$$y=\begin{cases} x^2+x+1 & (x\le 0) \\ x^2-x+1 & (x>0) \end{cases}$$

程序代码如下：

```
Define axz2()=
Prgm
Request "请输入x？ ",x
If x≤0 Then
x^2+x+1→y
Else
x^2-x+1→y
EndIf
Disp "y=",y
EndPrgm
```

四、 关系表达式与逻辑表达式

1．关系表达式和关系运算符

前面在 IF 语句中所用到的“条件”，例如“$n\le 0$”或“$a>0$”等，就是关系表达式。所谓关系表达式就是用一个关系运算符连接两个数值（或变量、计算表达式）。

TI-basic 语言中有六种关系运算符，分别用按键[=]以及组合按键[ctrl]+[=]调用。

关系运算符	含义	计算机键盘输入方式
=	等于	=
>	大于	>
<	小于	<
≠	不等于	/=
≥	大于或等于	>=
≤	小于或等于	<=

关系表达式会返回一个结果，“真”或者“假”。当给定的条件满足（成立）时，关系表达式的结果为“真”，否则为“假”。

例如，当：A=3，B=5，X=－3.5，Y=6.3 时，关系表达式的结果如下表所示：

关系表达式（条件）	条件满足否	关系表达式的结果
$A>B$	不满足	假
$X\geq Y$	不满足	假
$X<8.3$	满足	真
$X\neq 0$	不满足	假
$A-B>X-Y$	不满足	假
$A*B>13.6$	满足	真

2．逻辑表达式和逻辑运算符

一个简单的条件，可以用一个关系表达式来表示，但如果要指定的是一个由几个简单条件组成的复合条件，如“$0<a\leq 10$”、“$B>0$或$B<-10$”，就要把几个关系表达式联结组成一个逻辑表达式，如上述两个“复合条件”可以用下面两个式子表示：

$a > 0$ and $a \le 10$

$B > 0$ or $B < -10$

这就是两个逻辑表达式。

TI-basic 语言中，常用的逻辑运算符有三个：

逻辑运算符	名称	逻辑表达式运算规则
and	与	如果其前后的两个简单条件都满足，则认为满足此复合条件
or	或	如果在其前后两个简单的条件之一满足，则认为满足此复合条件
not	非	对写在其后的条件取“反”

假设 C 和 D 代表两个关系表达式，只有当 C 和 D 都为真时，C and D 的值才为真。只要 C 或 D 之一为真，则 C or D 的值为真。若 C 为真，则 not(C)的值为假。如下表。

C	D	C and D	C or D	not(C)	not(D)
真	真	真	真	假	假
真	假	假	真	假	真
假	真	假	真	真	假
假	假	假	假	真	真

【编程案例 3】计算运费。设货物运输价格 P（元）与运输距离 S（千米）有关，按以下公式定价（P 为每吨・千米的运价）。

$$P = \begin{cases} 20 & (S < 100) \\ 17.5 & (100 \le S < 200) \\ 15 & (200 \le S < 300) \\ 12 & (S \ge 300) \end{cases}$$

输入运距 S（千米）和货物重量 W（吨），要求计算出总运费。

COST（总运费）$= P \cdot W \cdot S$

验证数据：

运距 S （千米）	货物重量 W （吨）	运输价格 P （元/吨・千米）	总运费 （元）
80	8	20	12,800
160	9	17.5	25,200
220	5	15	16,500
300	4	12	14,400

程序代码如下：

```
Define axz3()=
Prgm
Request "运距(KM)? ",s
Request "货物重量(T)? ",w
If s<100 Then
20→p
ElseIf s≥100 and s<200 Then
17.5→p
ElseIf s≥200 and s<300 Then
15→p
Else
12→p
EndIf
p·w·s→cost
Disp "运输价格=",p
Disp "总运价=",cost
EndPrgm
```

课后练习题：

【编程练习 1】铁路托运行李，从甲地到乙地，按规定每张客票托运行李不超过 50 公斤时，每公斤 0.13 元，如果超过 50 公斤，超过的部分按每公斤 0.20 元计算。编写一个程序，输入行李的重量 W（单位：公斤），计算相应的运费 C（单位：元）。

提示：按照题意，运费的计算公式可写为：

$$C=\begin{cases}0.13\cdot W & (W\le 50)\\ 0.13\times 50+0.20\cdot(W-50) & (W>50)\end{cases}$$

【编程练习 2】有一函数：

$$y=\begin{cases}+1 & (x>0)\\ 0 & (x=0)\\ -1 & (x<0)\end{cases}$$

试编写程序，输入任意 x 值，输出 y 值。

【编程练习 3】给定一个年份，判断它是否闰年。闰年的条件是：能被 4 整除但不能被 100 整除，或者能被 100 整除且能被 400 整除。（例如 1989 年、1900 年不是闰年，2000 年、2012 年是闰年）。

【编程练习 4】目前，个人所得税的起征点为 3500 元，试根据个人所得税税率表，编程计算个人所得税。

级数	级距	税率(%)	速算扣除数
1	不超过 1,500 元的	3	0
2	超过 1,500 元至 4,500 元的部分	10	105
3	超过 4,500 元至 9,000 元的部分	20	555
4	超过 9,000 元至 35,000 元的部分	25	1,005
5	超过 35,000 元至 55,000 元的部分	30	2,755
6	超过 55,000 元至 80,000 元的部分	35	5,505
7	超过 80,000 元的部分	45	13,505

提示：个人所得税计算方法有两种：

（1）逐个按级距及对应税率计算

如：月收入 10000 元，超过起征点 6500 元，其中：

1500 元部分：1500×3%=45 元；

1500 元至 4500 元部分：3000×10%=300 元；

4500 元至 6500 元部分：2000×20%=400 元。

共计应缴个人所得税：45+300+400=745 元。

（2）个人所得税=（月收入—3500）×税率—速算扣除数

如：月收入 10000 元，超过起征点 6500 元，对应税率为 20%，速算扣除数 555 元，则：

个人所得税：（10000—3500）×20%—555=745 元。

验证数据：

月收入（元）	应缴个人所得税（元）	税后所得（元）
4,800	39	4,761
6,500	195	6,305
10,000	745	9,255
30,000	5,620	24,380
50,000	11,195	38,805
70,000	17,770	52,230
100,000	29,920	70,080