

# 程序控制结构-选择结构

车万翔

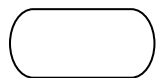
哈尔滨工业大学



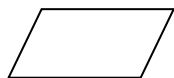
## ❖ 程序流程图

- 以简单的图形符号来表示问题的解决步骤，亦称为框图
- [重点] 流程图是问题求解的最基本、最重要的分析技术

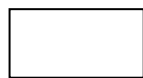
## ❖ 常用流程图图形符号



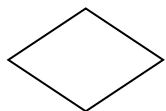
起止框



输入/输出框



处理框



判断框



流程线



连接符



# 程序流程图



哈爾濱工業大學  
HARBIN INSTITUTE OF TECHNOLOGY

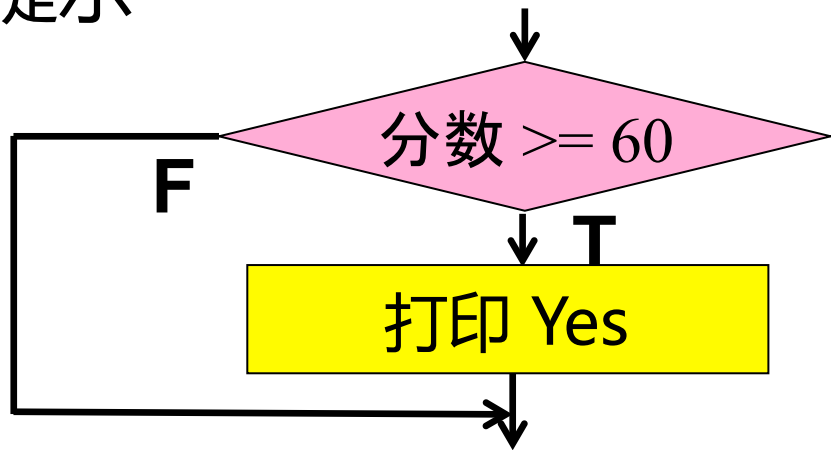
符 号	名称	含 义
	起止框	标准流程的开始与结束
	处理框	算法/程序要执行的处理操作
	判断框	判断条件是否成立
	文档框	以文件的方式输入/输出
	流程线	表示算法/程序执行的方向与顺序
	输入输出框	表示数据的输入/输出
	关联	同一流程图中从一个进程到另一个进程的交叉引用



# 程序流程图示例

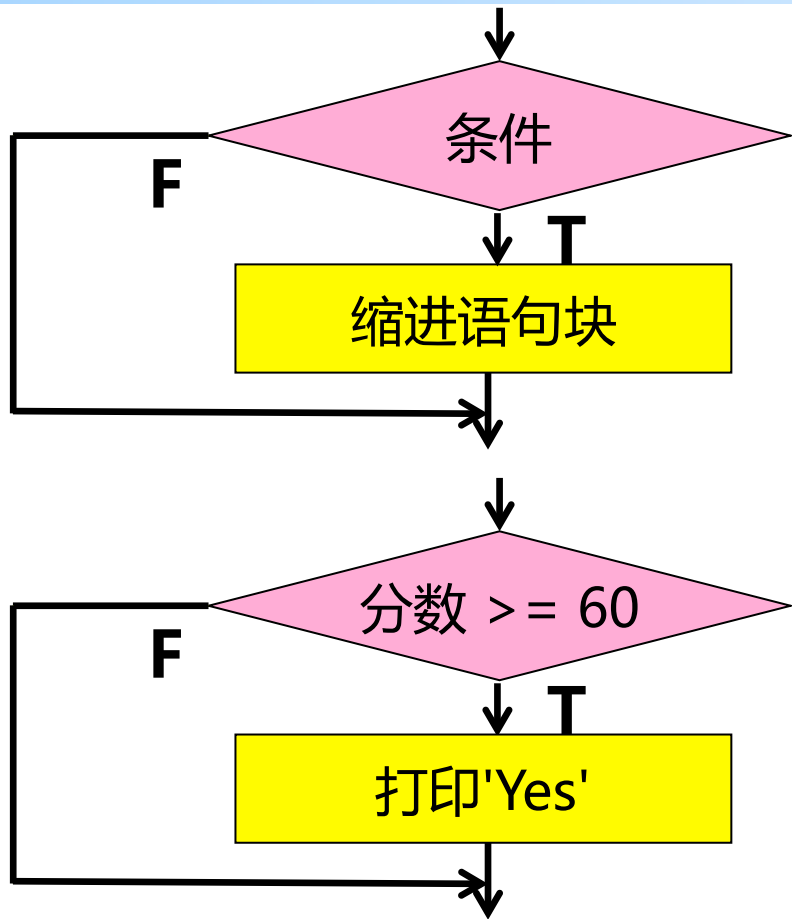


- ❖ 如果成绩合格，则打印相应提示
- ❖ 条件：合格  $\Leftrightarrow$  分数  $\geq 60$
- ❖ 动作：打印 Yes





# if 语句

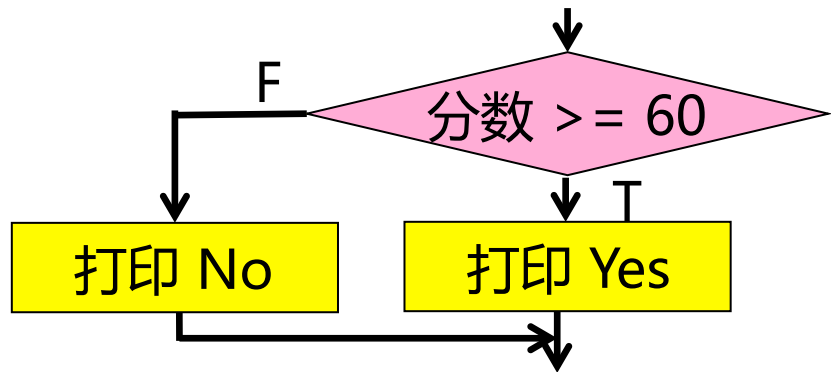
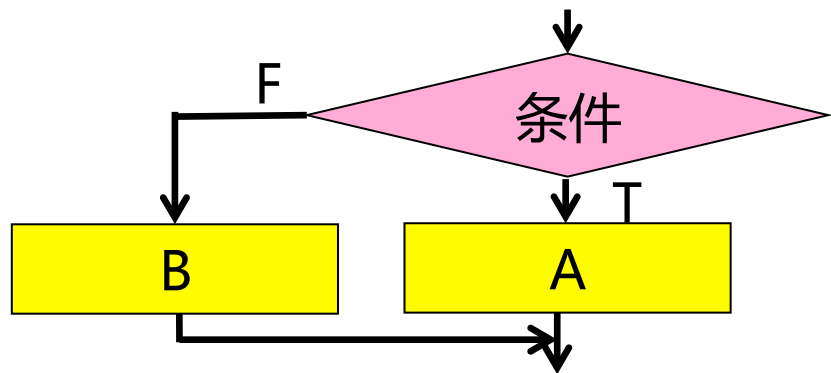


if 条件 :  
    缩进语句块  
其余语句

```
if score >= 60 :  
    print 'Yes'
```



# if-else 语句



if 条件 :  
    条件真缩进语句块

else:  
    条件假缩进语句块

```
if score >= 60 :  
    print 'Yes'  
else:  
    print 'No'
```



# if 语句-嵌套结构 (Nested)



哈爾濱工業大學  
HARBIN INSTITUTE OF TECHNOLOGY

❖ 如果是成绩合格的女生，则打印提示

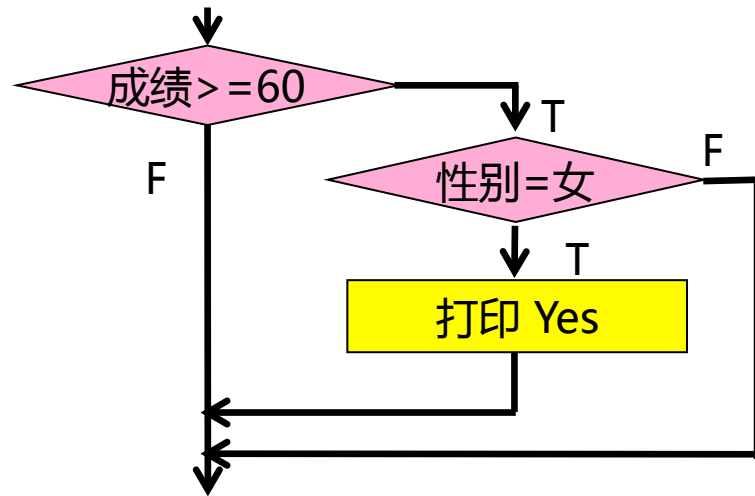
❖ 条件：成绩  $\geq 60$  且 性别=女

❖ 动作：打印 Yes

if score  $\geq 60$  :

    if gender == '女' :

        print 'Yes'



if score  $\geq 60$  and gender == '女' :

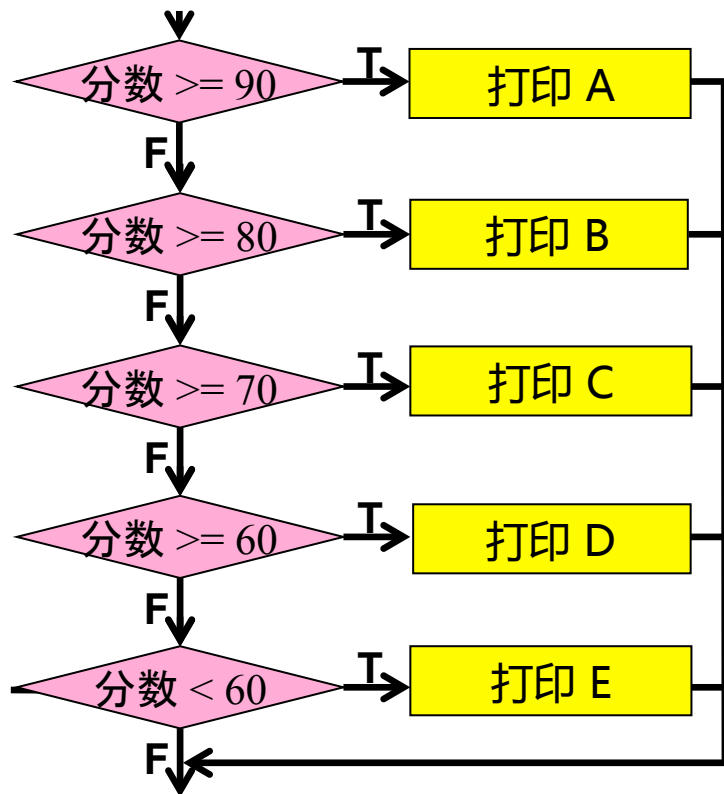
    print 'Yes'



# 多分支结构 ( Chained )



## ❖ 将考试分数转换为等级



```
7 score = 78
8
9 if score >= 90:
10     print 'A'
11 else:
12     if score >= 80:
13         print 'B'
14     else:
15         if score >= 70:
16             print 'C'
17         else:
18             if score >= 60:
19                 print 'D'
20             else:
21                 print 'E'
```





# 多分支结构 ( Chained )



```
7 score = 78
8
9 if score >= 90:
10     print 'A'
11 else:
12     if score >= 80:
13         print 'B'
14     else:
15         if score >= 70:
16             print 'C'
17         else:
18             if score >= 60:
19                 print 'D'
20             else:
21                 print 'E'
```



```
23 if score >= 90:
24     print 'A'
25 elif score >= 80:
26     print 'B'
27 elif score >= 70:
28     print 'C'
29 elif score >= 60:
30     print 'D'
31 else:
32     print 'E'
```

if-elif-else语句:

- elif 相当于 else: if , 但和第一个if条件并列
- if-elif-else 语句中有 else 条件时, else 条件放最后, 否则SyntaxError



# 示例：求一元二次方程的解



❖ 一元二次方程  $ax^2 + bx + c = 0$

❖ 解为：  
$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

```
8 import math
9
10 a = float(raw_input('Input a: '))
11 b = float(raw_input('Input b: '))
12 c = float(raw_input('Input c: '))
13
14 root = math.sqrt(b ** 2 - 4 * a * c)
15 s1 = (-b + root) / (2 * a)
16 s2 = (-b - root) / (2 * a)
17
18 print 'The solutions are: ', s1, s2
19 |
```

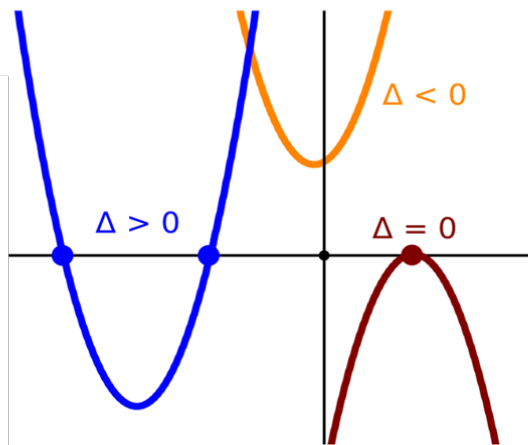


# 示例：求一元二次方程的解



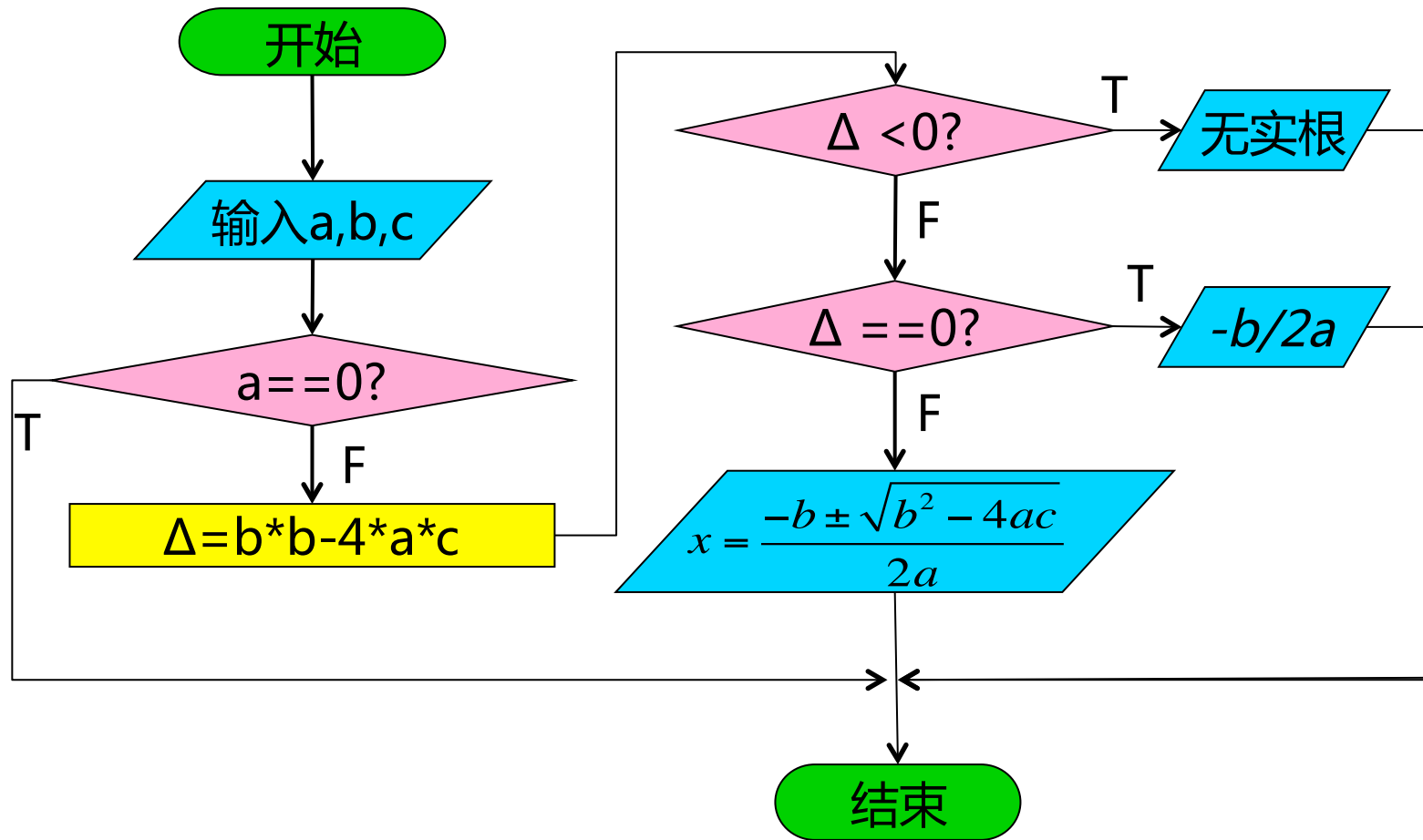
一元二次方程 $ax^2+bx+c=0$  ( $a \neq 0$ ) 的根与 $\Delta=b^2-4ac$ 有如下关系：

- ①当 $\Delta > 0$ 时，方程有两个不相等的两个实数根；
- ②当 $\Delta = 0$ 时，方程有两个相等的两个实数根；
- ③当 $\Delta < 0$ 时，方程无实数根。





# 示例：流程图





# 示例：源代码



```
8 import math
9
10 a = float(raw_input('Input a: '))
11 b = float(raw_input('Input b: '))
12 c = float(raw_input('Input c: '))
13
14 if a == 0:
15     print 'The equation is linear, not quadratic'
16 else:
17     delta = b ** 2 - 4 * a * c
18     if delta < 0:
19         print 'No real roots!'
20     elif delta == 0:
21         print 'Only one root is ', -b / (2 * a)
22     else:
23         root = math.sqrt(delta)
24         s1 = (-b + root) / (2 * a)
25         s2 = (-b - root) / (2 * a)
26
27         print 'Two distinct solutions are: ', s1, s2
28
```



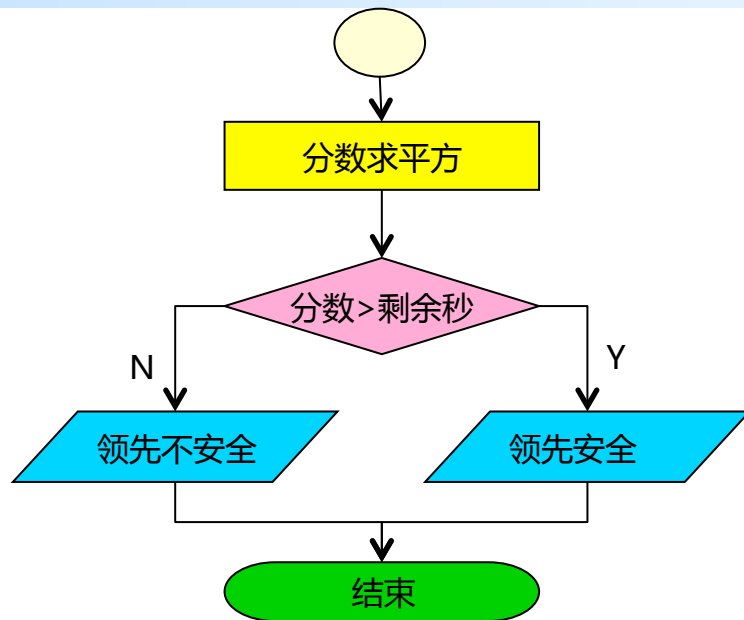
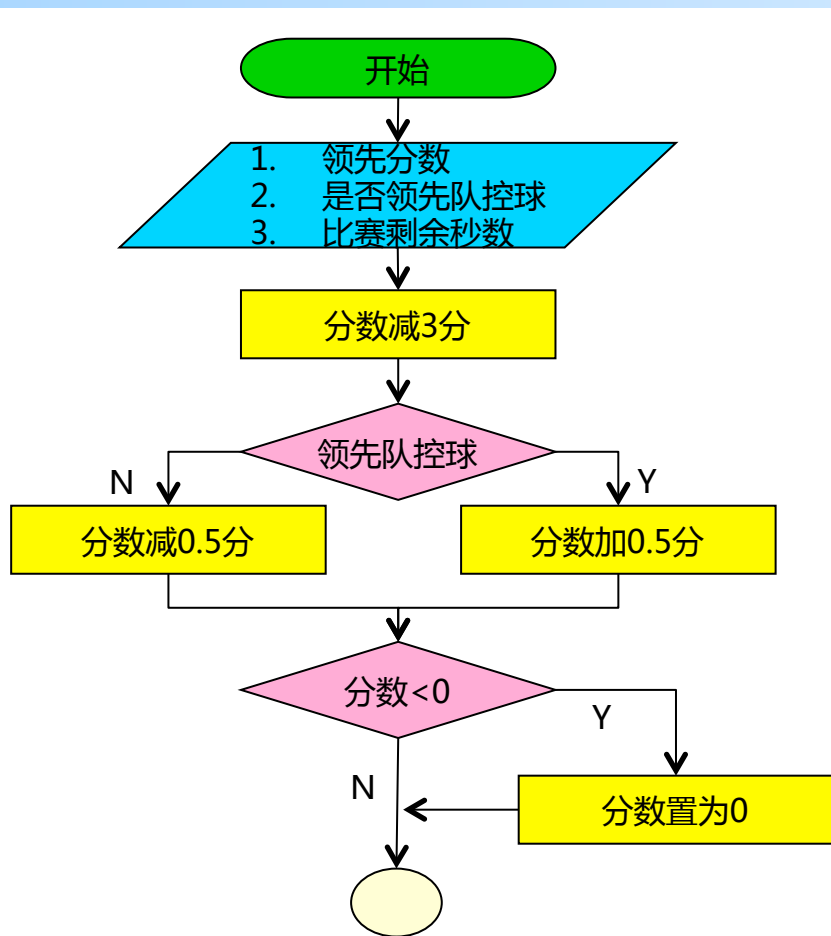
# 示例：篮球比赛领先多少才安全？



- ❖ 篮球比赛是高的分的比赛，领先优势可能很快被反超。作为观众，希望能在球赛即将结束时，就提早知道领先是否不可超越。体育作家Bill James发明了一种算法，用于判断领先是否“安全”
- ❖ 算法描述
  - 获取领先的分数
  - 减去三分
  - 如果目前是领先队控球，则加0.5；否则减0.5（数字小于0则变成0）
  - 计算平方后的结果
  - 如果得到的结果比当前比赛剩余时间的秒数大，则领先是“安全”的



# 示例：流程图





# 示例：代码



```
8 points = int(raw_input('Input the lead in points: '))
9 has_ball = raw_input('Does the lead team have the ball (Yes or No): ')
10 seconds = int(raw_input('Input the number of secondes remaining: '))
11
12 points -= 3
13
14 if has_ball == 'Yes':
15     points += 0.5
16 else:
17     points -= 0.5
18
19 if points < 0:
20     points = 0
21
22 points **= 2
23
24 if points > seconds:
25     print 'Lead is safe'
26 else:
27     print 'Lead is not safe'
28
29 |
```