第 3 章　语句与结构化程序设计

3.1　学 习 要 求

(1) 掌握程序的 3 种常见流程结构。

(2) 掌握 Python 分支结构程序和循环结构程序的一般设计方法。

3.2　知 识 要 点

1. Python 语句

Python 程序由 Python 语句组成,通常一行编写一个语句。例如:

```
print('Hello')
print('I am Python')
```

Python 语句可以没有结束符,不像 C 或 Java 那样在语句后面必须有分号(;)表示结束。当然,Python 程序中也可以根据习惯在语句后面使用分号(;)。

也可以把多个语句写在一行,此时就要在语句后面加上分号(;)表示结束。

把多个语句写在一行的例子。

```
print('Hello'); print('I am Python');
```

2. 缩进

缩进是指在代码行前面添加空格或 Tab,使程序更有层次、更有结构感,从而使程序更易读。

缩进是 Python 语言中表明程序框架的唯一手段。

在 Python 程序中,缩进不是任意的。平级的语句行(代码块)的缩进必须相同。

在 Python 中,1 个缩进=4 个空格。

3. 顺序结构

图 3-1 是一个顺序结构的流程图,它有一个入口、一个出口,依次执行语句 1 和语句 2。

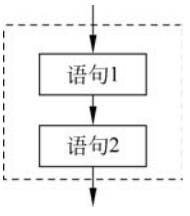

图 3-1　顺序结构

一般情况下,实现程序顺序结构的语句主要是赋值语句、内置的输入函数 input()和输出函数 print()。这些语句可以完成输入、计算、输出的基本功能。

【例 3-1】 求解三角形面积。

```
a,b,c = input("请输入三角形的 3 条边长: ").split(" ")      # 输入空格分开的 3 个值
a,b,c = int(a),int(b),int(c)  # input 函数默认输入字符串,这里转换为整型
s = (a + b + c)/2
area = (s * (s-a) * (s-b) * (s-c)) ** 0.5
print("面积",area)
```

4. 分支结构

分支结构就是按照给定条件有选择地执行程序中的语句。

在 Python 语言中,实现程序分支结构的语句有: if 语句(单分支)、if…else 语句(双分支)和 if…elif 语句(多分支)。

Python 中 if 语句的一般形式如下所示:

```
if condition_1:
    statement_block_1
elif condition_2:
    statement_block_2
else: statement_block_3
```

注意:

(1) 每个条件后面要使用冒号":"表示接下来是满足条件后要执行的语句块。

(2) 使用缩进来划分语句块,相同缩进数的语句在一起组成一个语句块。

(3) 在 Python 中没有 switch…case 语句。

【例 3-2】 输入两个整数 a 和 b,按从小到大的顺序输出这两个数(单分支)。

```
a = eval(input("a = "))
b = eval(input("b = "))
if a > b:
    a,b = b,a
print(a,b)
```

【例 3-3】 输入一个年份 year,判断是否为闰年(双分支)。

```
year = eval(input("输入年份: "))   # 也可用 int()函数
if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
    print(year,": 闰年")
else:
    print(year,": 非闰年")
```

【例 3-4】 求下面分段函数的结果(多分支)。

$$y=\begin{cases}|x| & (x<0)\\ e^{x}\cos x & (0\leqslant x<15)\\ x^{5} & (15\leqslant x<30)\\ (7+9x)\ln x & (x\geqslant 30)\end{cases}$$

```
from math import *                    # 导入数学模块 math
x = eval(input("请输入 x: "))
if x<0 :
```

```
    y = abs(x)
elif x < 15 :
    y = exp(x) * cos(x)                # exp(x)在 math 中
elif x < 30 :
    y = pow(x,5)                       # 等价于 x ** 5
else :
    y = (7 + 9 * x) * log(x)           # log(x)在 math 中
print("y = ", y)
```

5. 循环结构

循环结构是一种让指定的代码块重复执行的有效机制。Python 可以使用循环使得在满足“预设条件”下,重复执行一段语句块。构造循环结构有两个要素:一个是循环体,即重复执行的语句和代码;另一个是循环条件,即重复执行代码所要满足的条件。为了能够适应不同场合的需求,Python 用 while 和 for 关键字来构造两种不同的循环结构,即表达两种不同形式的循环条件。

while 语句用于实现当型循环结构,在给定的判断条件表达式为 True 时执行循环体,否则退出循环体,如图 3-2 所示。其特点是先判断,后执行。

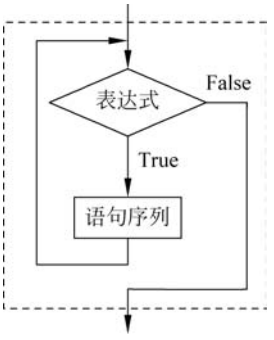

图 3-2 当型循环结构

语法格式:

```
while <表达式> :
    <语句序列>
```

【例 3-5】 求自然数 1～100 之和。

```
i = 1
sum = 0
while i <= 100 :
    sum += i                  # 等价于 sum = sum + i
    i += 1
print("sum = ", sum)
```

for 语句是一种遍历型循环,也就是说在循环的起始位置需要设置一个遍历范围或遍历的数据集合,在 for 循环的执行过程中,它会将该范围或集合中的数据带入到循环体中逐个执行一遍,直到所有的数据都尝试过为止。

语法格式:

```
for <变量> in <可迭代容器> :
    <语句序列>
```

其中,<变量>可以扩展为变量表,变量与变量之间用“,”分开。<可迭代容器>可以是序列、迭代器或其他支持迭代的对象。

【例 3-6】 求 Fibonacci 数列的前 20 项,并输出。

分析:Fibonacci 数列为 0,1,1,2,3,5,8,13,21,…,即 $f(0)=0$,$f(1)=1$,$f(n)=f(n-1)+f(n-2)(n\geqslant 2)$。

```
a, b = 0, 1
for i in range(20) :
```

```
    if (i + 1) % 5 != 0 :
        print(a, end = '\t')
    else :
        print(a, end = '\n')
    a, b = b, a + b
```

有时需要在循环体中提前跳出循环，或者在某种条件满足时，不执行循环体中的某些语句而立即从头开始新的一轮循环，这时就要用到循环控制语句 break、continue 和 pass 语句。

break 语句用在循环体内，迫使所在循环立即终止，即跳出所在循环体，继续执行循环结构后面的语句。

continue 语句在循环语句中强行提前结束本次循环，而不是终止循环。

【例 3-7】 求 1～100 的全部奇数之和。

```
x = y = 0
while True:
    x += 1
    if  x % 2 == 0:continue
    elif x > 100:break
    else:y += x
print("y = ",y)
```

pass 语句是一个空语句，它不做任何操作，代表一个空操作，在特别的时候用来保证格式或语义的完整性。例如下面的循环语句：

```
for i in range(5):
    pass
```

该语句的确会循环 5 次，但是除了循环本身之外，它什么也没做。


视频讲解

3.3 应 用 举 例

【例 3-8】 编程打印如下所示的三角形图案。

(1) 示例 1。

```
*
**
***
****
*****
```

(2) 示例 2。

```
    *
   **
  ***
 ****
*****
```

(3) 示例 3。

```
    *
   ***
  *****
 *******
*********
```

(4) 示例 4。

```
    *
   * *
  * * *
 * * * *
* * * * *
```

(5) 示例 5。

```
*
* *
*   *
*     *
* * * * *
```

程序如下：

(1) 示例 1。

```
for i in range(1,6):
    print(i * "*")  # 在同一行打印 i 个 *
```

(2) 示例 2。

```
for i in range(1,6):
    print((5 - i) * " ",i * "*")
```

(3) 示例 3。

```
for i in range(1,6):
    print((5 - i) * " ",(2 * i - 1) * "*")
```

(4) 示例 4。

```
for i in range(1,6):
    print((5 - i) * " ",i * "* ")
```

(5) 示例 5。

```
for i in range (1,5):
    for j in range (1,i + 1):
        if j == 1 or j == i:
            print("*",end = "")
        else:
            print(" ",end = "")
    print()
print(5 * "*")
```

【例 3-9】 从键盘输入一个 3 位整数，分离出它的个位、十位和百位并在屏幕上用一条 print 语句格式化输出两次(分别用%d 和{}.format)。

```
x = int(input('请输入一个 3 位整数: '))
a = x % 10
b = x//10 % 10
c = x//100
print('个位 = %d,十位 = %d,百位 = %d' % (a,b,c))
print('个位 = {},十位 = {},百位 = {}'.format(a,b,c))
```

【例 3-10】 从键盘输入一个字符 ch，判断并输出它是英文字母(输出用%s 格式)、数字或其他字符(输出用{}.format)。

```
ch = input('请输入一个字符: ')
if ch >= 'a' and ch <= 'z' or ch >= 'A' and ch <= 'Z':
    print('%s 是英文字母' % ch)
elif ch >= '0' and ch <= '9':
    print('{}是数字'.format(ch))
else:
    print('{}是其他字符'.format(ch))
```

【例 3-11】 输入三角形的 3 条边长，求三角形的面积(先判断 3 条边是否能构成三角形，不引入数学库)。

```
a,b,c = eval(input('用英文逗号分隔输入 a,b,c = '))
if a + b > c and a + c > b and b + c > a:
    p = (a + b + c) / 2
    area = (p * (p - a) * (p - b) * (p - c)) ** 0.5
    print('面积: %f' % (area))
else:
    print('不能构成三角形')
```

习　　题

1. 编写程序，接收用户从键盘上输入的 3 个整数，求出其中的最小值并输出在屏幕上。

2. 编写程序，接收用户从键盘输入的一个 1～7 的整数，该整数表示一个星期中的第几天，在屏幕上输出对应的英文单词(提示：1 表示星期一，7 表示星期日)。

3. 编写程序，输出 1～100 中所有 3 的倍数，并规定一行输出 5 个数。

4. 编写程序，打印如下倒三角形。

```
* * * * * * *
  * * * * *
    * * *
      *
```

5. 编写程序，打印九九乘法口诀表(提示：为了让算式对齐显示，使用 format()方法格式化输出字符串)。

6. 编写程序，找出 1000 以内的“完数”并输出，同时输出找到的完数个数。所谓“完数”就是数本身等于其各因子之和的数，如 6＝1＋2＋3。