简单分支

礼 欣

北京理工大学

PM2.5指数分级例子

程序功能的IPO模式描述为：

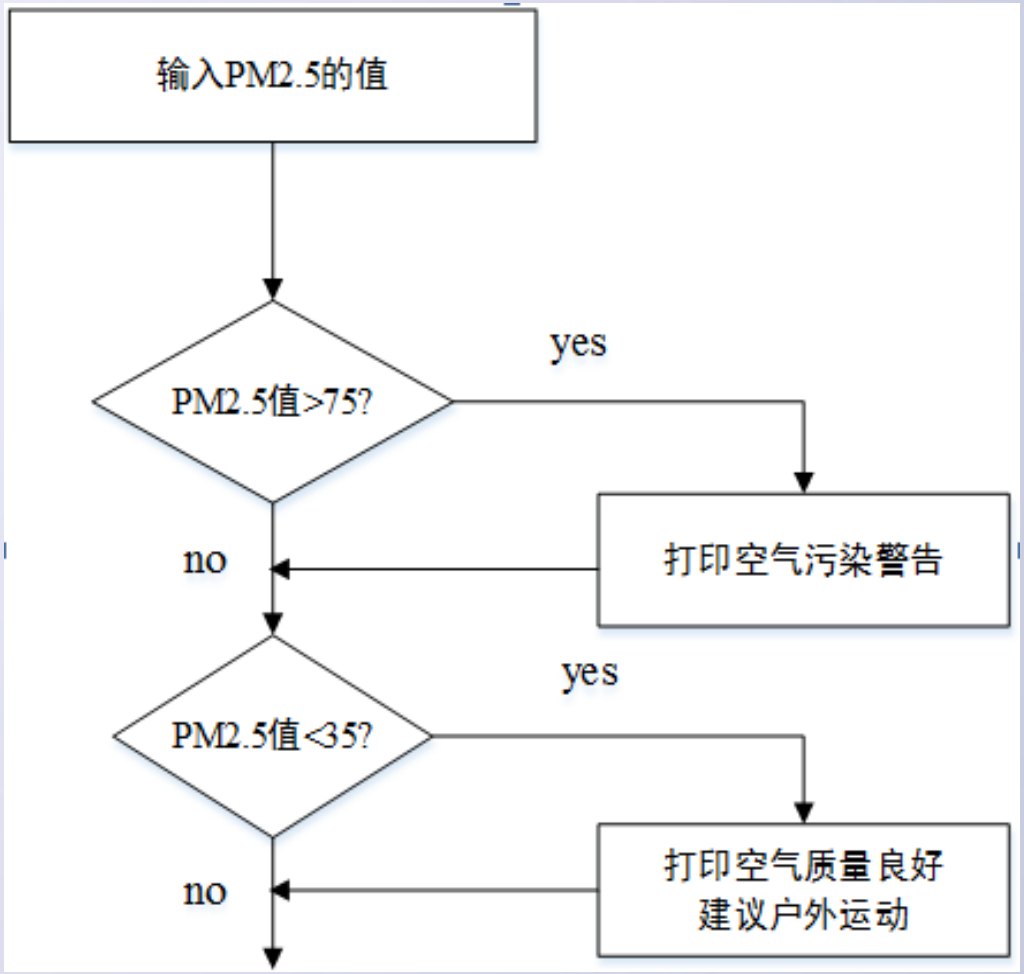
- 输入：接受外部输入PM2.5值
- 处理：空气质量分级算法
- 输出：打印空气质量提醒

伪代码如下：

```
if PM2.5值 > 75
        打印空气污染警告
if PM2.5值 < 35
        打印空气质量优，建议户外运动
```

PM2.5	空气质量等级
0-35	优
35-75	良
75-115	轻度污染
115-150	中度污染
150-250	重度污染
250-500	严重污染

PM2.5指数分级例子-流程图



python 语言程序设计

pm25.py

```
# pm25.py
# 空气质量提醒

def main():
    PM = eval(input("What  is today' s PM2.5? "))
    # 打印相应提醒
    if PM  > 75:
        print("Unhealthy. Be careful!")
    if PM  < 35:
        print("Good.  Go running!")
main()
```

python 语言程序设计

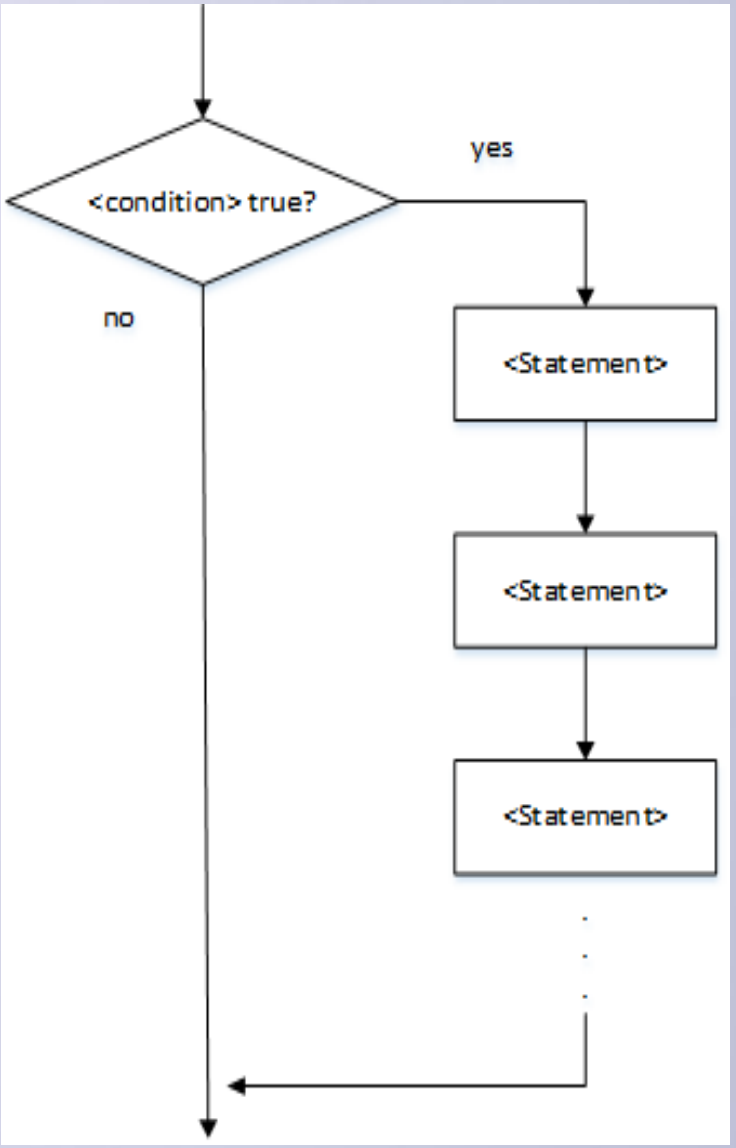
if语句格式

- 语句格式如下

  if <condition>:

  <body>

  - 其中<condition>是条件表达式，<body>是一个或多个语个或多个语句序列。
- 先判断<condition>条件：
  - true，则执行<body>，再转向下一条语句；
  - false，则直接跳过<body>，转向下一条语句；



简单条件构造

- 简单条件基本形式

<expr> <relop> <expr>

  - <relop>是关系操作符<, <=, ==, >=, >, !=
  - 使用“=”表示赋值语句，使用“==”表示等于
  - 除数字外，字符或字符串也可以按照字典顺序用于条件比较
  - <condition>是布尔表达式，为bool类型，布尔值的真和假以字符True和False表示

简单条件构造

示例：

```
>>> 3 < 4
True
>>> 3 * 4 < 3 + 4
False
>>> "hello" == "hello"
True
>>> "hello" < "hello"
False
>>> "Hello" < "hello"
True
```

python 语言程序设计

二次方程求解-例（版本1）

```
# quadratic.py
# 计算二次方程的实数根程序
# 此程序在方程没有实根的情况下报错
import math
def main():
    print("This program find the real solutions to a quadratic\n")
    a, b, c = eval(input("Please enter the coefficients (a, b, c): "))
    delta = b * b - 4 * a * c
    if delta >= 0:
        delta = math.sqrt(delta)
        root1 = (-b + delta) / (2 * a)
        root2 = (-b - delta) / (2 * a)
        print("\nThe solutions are:", root1, root2 )
main()
```

python 语言程序设计

版本1执行

```
>>>
This program finds the real solutions to a  quadratic

Please enter the coefficients(a, b,  c):1,2,3
Traceback (most recent call last):
  File "C:\Python34\Scripts\quadratic.py", line 12, in <module>
    main()
  File "C:\Python34\Scripts\quadratic.py", line 8, in main
    delta  = math.sqrt(b  *  b  - 4  *  a  *  c)
ValueError: math domain error
>>>
```

- 因为当$b^2 - 4ac$小于0时，程序将试图对一个负数开根号，程序将报错
- 通过引入决策判断来改进版本1

二次方程求解（版本2）

```
# quadratic2.py
import math
def main():
    print("This program find the real solutions to a quadratic\n")
    a, b, c = eval(input("Please enter the coefficients (a, b, c): "))
    delta = b * b - 4 * a * c
    if delta >= 0:
        delta = math.sqrt(delta)
        root1 = (-b + delta) / (2 * a)
        root2 = (-b - delta) / (2 * a)
        print("\nThe  solutions  are:",  root1,  root2  )
main()
```

- 改进后程序首先判断delta值，只有非负情况时，程序才进行开方求解。
- Delta为负时，程序不调用sqrt()函数

版本2执行

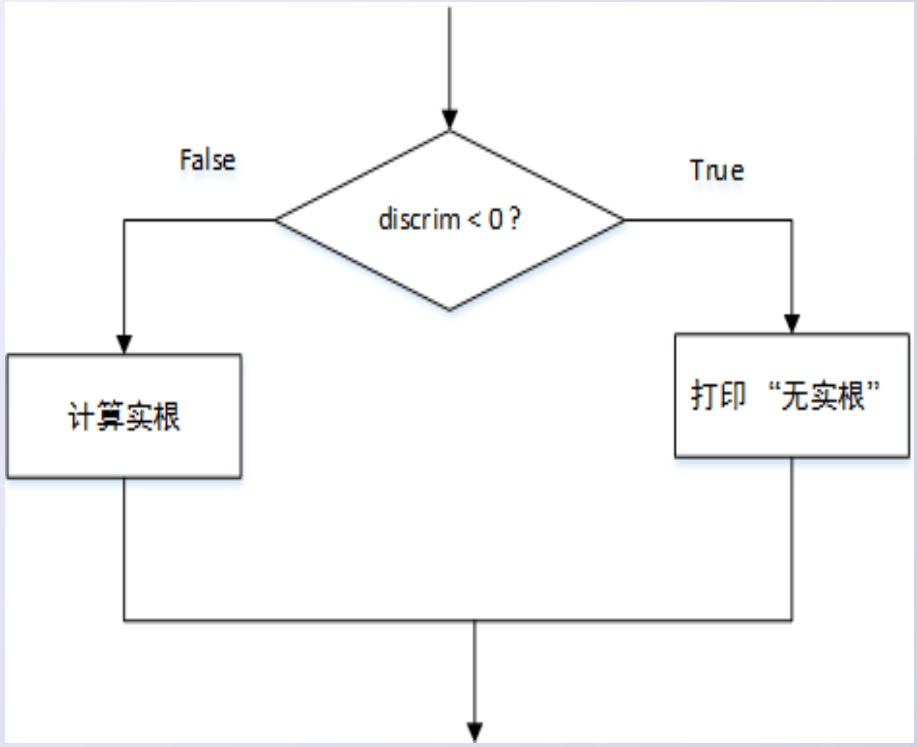
```
>>>
This program find the real solutions to a quadratic

Please enter the coefficients (a, b, c): 1,2,3
>>>
```

- 当$b^2 - 4ac$小于零，程序将简单地跳过计算
- 这个程序的缺点在于它没有给用户任何的出错反馈，通过在程序尾部添加另一个简单决策来实现消息打印

```
if delta < 0:
    print("The equation has no real roots!")
```

- 以上改进的确解决了出错的问题，但是仍然不够完美。

二分支决策



python 语言程序设计

二分支决策

- 二分支语法结构如下：

```
if <condition>:
    <statements>
else:
    <statements>
```

- Python解释器首先评估<condition>
  - 如果<condition>是真的，if下面的语句被执行
  - 如果<condition>是假的，else下面的语句被执行。

二次方程求解（版本3）

```
# quadratic3.py
import math
def main():
    print("This program finds the real solutions to a quadratic\n")
    a, b, c = eval(input("Please enter the coefficients (a, b, c): "))
    delta = b * b - 4 * a * c
    if delta < 0:
        print("\nThe equation has no real roots!")
    else:
        delta = math.sqrt(delta)
        root1 = (-b + delta) / (2 * a)
        root2 = (-b - delta) / (2 * a)
        print("\nThe solutions are:", root1, root2)
main()
```

python 语言程序设计

版本3执行

```
>>>
This program finds the real solutions to a quadratic

Please enter the coefficients (a, b, c): 1,2,3

The equation has no real roots!
```

```
>>>
This program finds the real solutions to a quadratic

Please enter the coefficients (a, b, c): 2,4,1

The solutions are: -0.2928932188134524 -1.7071067811865475
```

python 语言程序设计