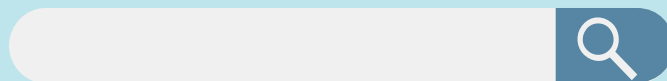


第四章

VB 流程控制





一、IF语句



二、Select Case语句



三、IIF函数



四、For循环





五、Do循环



六、退出循环



七、练习


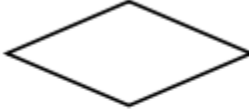
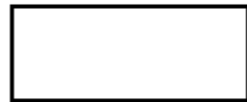





一、If语句



VB流程图描述程序逻辑

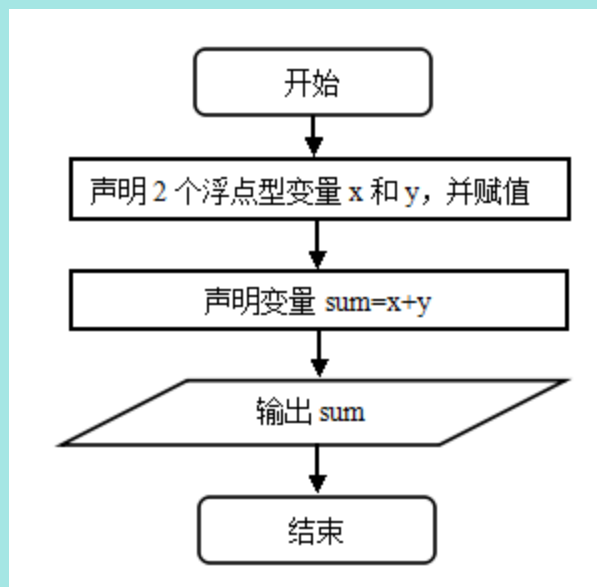
流程图是程序步骤的图形化表示方法。流程图中包括如下符号：

图形	意义	图形	意义
	程序开始或结束		判断和分支
	计算或处理		连接符
	输入或输出		流程线

以上图形中，流程线用来连接相邻的两个步骤；每一个程序都有且仅有一个开始和结束。

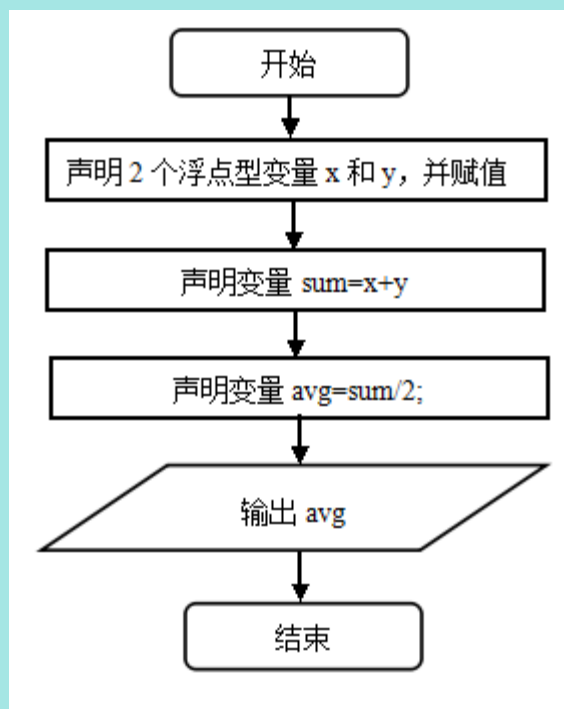
1.1用流程图描述程序

VB以下流程图描述的是，求2个浮点数的和，后边是用C#实现：



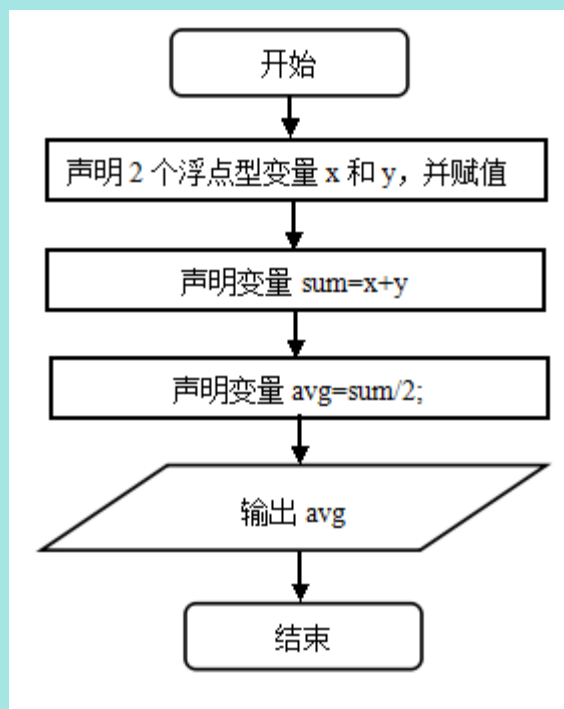
1.1用流程图描述程序

请参考下面的流程图，完成代码。



1.1用流程图描述程序

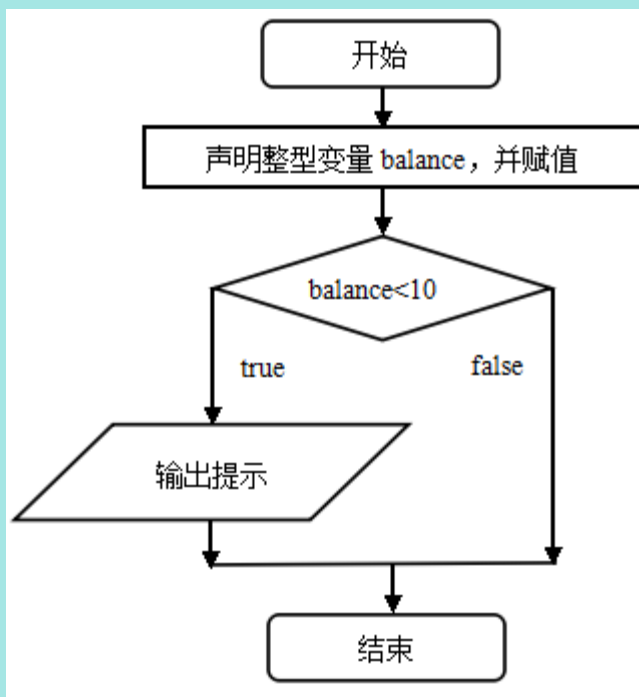
请参考下面的流程图，完成代码。



1.2判断和分支



请走到岔路口，需要选择方向。编写程序也会遇到判断和分支。请看下面的流程图，判断手机账户余额是否不足10元，如果不足给出提示：



1.2判断和分支



这个程序在 “balance<10” 这个步骤出现了分支，“balance<10” 被称为判断（ Boolean 类型），当判断为 True 时，执行左边的分支，输出提示；当判断为 False 时，执行右边的分支，不输出任何内容。在VB中，这个结构成为条件结构，通常用 If 关键字来实现：

```
' 余额
Dim balance As Double = 9.78
' 判断
If balance < 10 Then
    Console.WriteLine("余额不足，请充值")
End If
```

当 If 右边的判断为 True 的时候，会执行分支一；如果判断为 False，则不会执行分支一。

1.2判断和分支

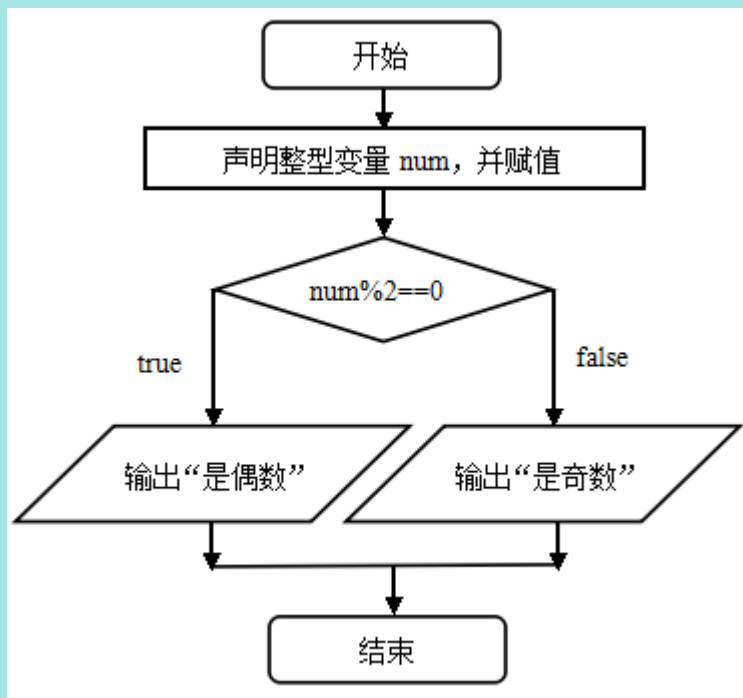


练习：程序判断考试分数是否高于95分，如果高于95分，给予奖励。

1.3 If...Else 条件结构



条件结构可以有2个分支，比如下面的流程图，判断一个整数是奇数还是偶数

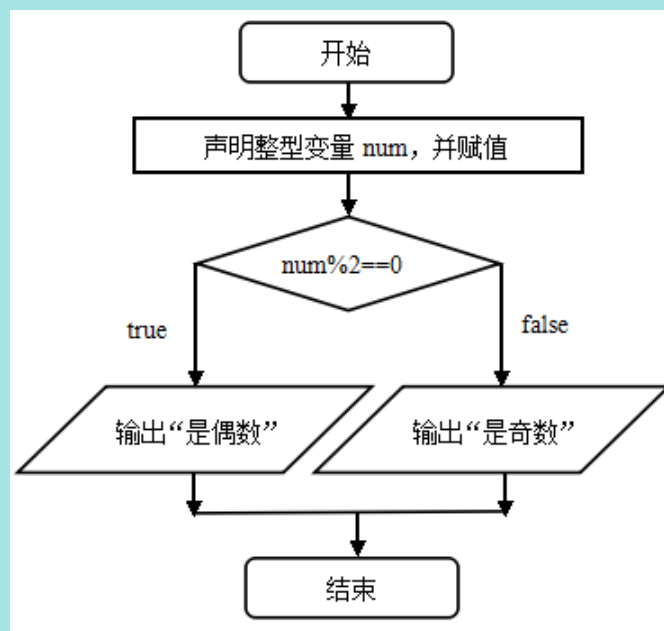


1.3 If...Else 条件结构



其中，条件为 true 时执行的分支写在 if() 后面的 {} 中；条件为 false 时执行的分支写在 else 后面的 {} 中。

这个由 if 和 else 组成的结构是条件结构的一种，它的基本逻辑是：当条件为 true 时，执行分支1，否则，执行分支2。这段程序中，变量 num 能够被2整除时会执行分支1，不能被2整除时执行分支2。



1.3 If...Else 条件结构



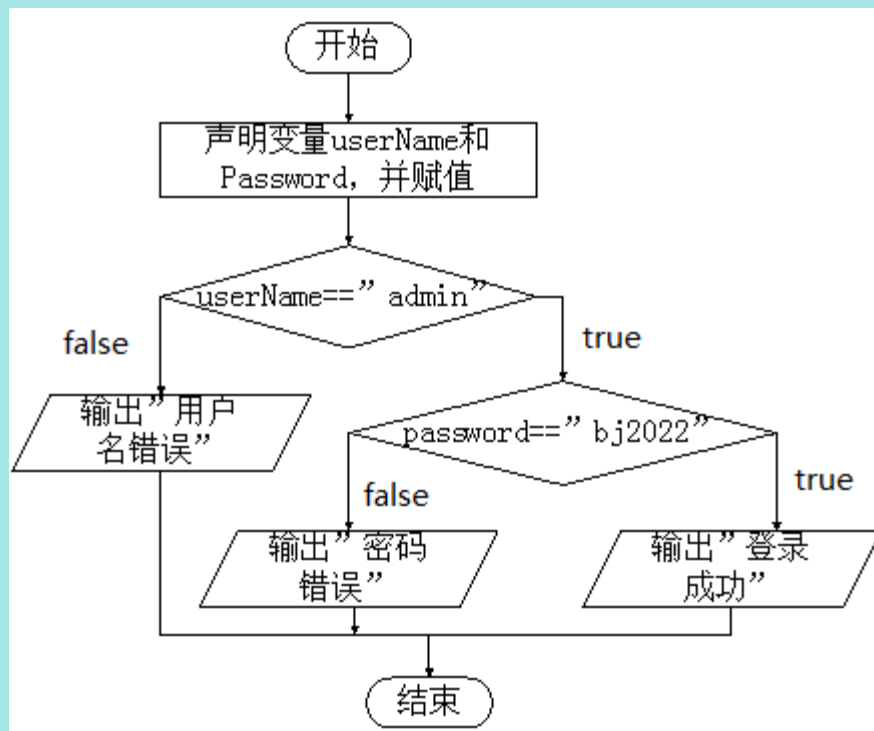
练习：根据年龄判断是否为成年人。18岁时成年

这个程序用 2 个 double 类型变量，分别记录了苹果六手机的售价和本月的是发工资，缺少一个 if...else 结构的判断，请补充这个条件判断，如果本月工资够买手机的，输出“这月工资够买手机！”否则输出“这月工资不够买手机！”

1.4嵌套If结构



程序开发中，往往需要先判断一个条件是否成立，再判断另一个条件。比如下面的例子：验证账号是否为“admin”，如果不是则提示错误；如果是，则验证密码是否为“bj2022”：



1.4嵌套If结构



练习：在我国，年满22岁的男性和年满20岁的女性可以结婚。请编写程序，使程序能够完成“是否达到法定婚龄”的验证。

1.4多重If结构



如果，有多个条件，其中只有一个成立，应该怎样判断呢？比如小明考试，90分以上奖励一台手机，80-89分奖励一个滑板，60-79分奖励一个烧饼，60以下没有奖励，0-100的分数被划分为4档，需要做3次条件判断。如果用上一节学习的嵌套if结构完成，3层if的嵌套会使程序变得很复杂。本节我们使用多重 if 结构实现，它的基本语法为：

```
if(条件1)
{
    分支1
}
else if(条件2)
{
    分支2
}
...
else
{
    分支n
}
```


二、Select Case语句

思考一个问题：

客户叫关羽吗？如果是。。。完成某个人物

客户叫张飞吗？如果是。。。完成某个人物

客户叫赵云吗？如果是。。。完成某个人物

客户叫黄忠吗？如果是。。。完成某个人物

客户叫马超吗？如果是。。。完成某个人物

我们可以写一堆If。。。Then来完成这个需求

但是代码太多不美观，效率也不高，

这时可以使用Select Case 语句

二、Select Case语句



```
Dim name As String = "关羽"
Dim skill As String
Select Case name
    Case "关羽"
        skill = "单刀咆哮"
    Case "张飞"
        skill = "变大猩猩"
    Case "赵云"
        skill = "枪尖在燃烧"
    Case "黄忠"
        skill = "意大利炮"
    Case "马超"
        skill = "没出这英雄"
    Case Else
        skill = "啦啦啦"
End Select
Console.WriteLine(skill)
```

三、IIF函数



根据[表达式](#)的值，来返回两部分中的其中一个。

```
Iif(skill = "啦啦啦", 1, 0)
```

四、For循环



```
For n = 1 To 5
    Console.WriteLine("第" & n & "项")
Next
For n = 4 To 62 Step 5
    Console.WriteLine("第" & n & "项")
Next
```

循环一定的次数，从4到62，步长是5
步长也可以是负数，来进行反向循环

```
For n = 4 To 62 Step -5
    Console.WriteLine("第" & n & "项")
Next
```

For Each ... Next 循环，用到对象的时候再讲

五、Do循环



```
Dim random As New Random()  
Dim randomNumber As Integer  
' 一直循环直到randomNumber=10  
Do Until randomNumber = 10  
    randomNumber = random.Next(25)  
    Console.WriteLine("随机数" & randomNumber)  
Loop  
' 当randomNumber小于15的时候才循环  
Do While randomNumber < 15  
    randomNumber = random.Next(25)  
    Console.WriteLine("随机数" & randomNumber)  
Loop
```

五、Do循环



```
Dim n1 As Integer = 0
Do While n1 < 3
    n1 += 1
Loop
Do
    n1 += 1
Loop While n1 < 3
```

初看起来，好像是While $n < 3$ 移动了位置，你可能觉得这两个循环是等效的，但他们有一些细微的差别，假设执行这两个Do循环时 n 大于3（比如 n 是4），则第一个循环根本不会执行，而第二个循环会执行一次。当执行Loop While $n < 3$ 时，循环就退出了，即使 n 小于3第二个循环也会执行一次。

五、Do 循环



```
Dim n1 As Integer = 0
Do Until n1 < 3
    n1 += 1
Loop
Do
    n1 += 1
Loop Until n1 < 3
```

同样，这两个循环看起来也是相同的，但他们也有细微的差别。假设n为3，则第一个循环不会执行。因为n已经满足退出循环的条件。然而第二个循环会执行一次。当第一次执行Loop Until n = 3时，n为4，接着返回循环的开头。并把n递增为5，依次类推。事实上，这是个死循环，永远不会停止。

使用Loop While和Loop Until循环时，不管怎样，都至少执行一次循环。一般来说，最好使用Do While 或Do Until 而不使用Loop While和Loop Until

六、退出循环



Exit For 退出For循环。Continue For 退出本次For循环
Exit Do 退出Do循环。Continue Do 退出本次Do循环

七、练习



1. 使用Do While循环 打印10次加油

2. 使用For循环求1-10的和。

Tips：我们曾经循环打印过 1-10 的整数，如果声明一个变量，在打印的过程中，把每一个数字都累加到变量里面，就可以得到 1-10 的和。求和算法的基本步骤：

①声明变量 sum 用来存储“和”。sum 的初始值赋为 0。

②循环，用变量 x 访问每一个数字，用 $sum += x$ 的方式累加。（即 $sum = sum + x$ ；）

③循环结束后，sum 中存储的即是和。

3. 使用代码循环输出 1-9 的整数，并使得 3 和 8 不会被打印。

4. 用代码打印右边三角形

5. 用代码打印右边图形

```
C:\windows\system32\cmd.exe
1
12
123
1234
12345
123456
1234567
请按任意键继续...
```

```
C:\windows\system32\cmd.exe
0....0
.0...0.
..0.0..
...0...
..0.0..
.0...0.
0....0
请按任意键继续...
```