第四周：程序的控制结构

4.1 程序的分支结构

1.单分支结构

```
    if<条件>:
        <语句块>
```

2.二分支结构

```
if <condition>:
   <yujukuai1>
else:
   <yujukuai2>
```

还有一种紧凑表达式：

```
<表达式 1>if<条件> else<表达式 2>
```

例如：

```
guess =eval(input())
print("猜{}了".format("对" if guess ==99 else "错"))
```

注意：紧凑形式只支持表达式，不支持其他类型语句（例如赋值语句）

3.多分支结构

例程：

```
score = eval(input())
if ((score >=90) and (score<=100)):
      grade = "A"
elif ((score >=80) and (score<90)):
      grade = "B"
elif ((score >=70) and (score<80)):
      grade = "C"
elif ((score >=60) and (score<70)):
      grade = "D"
elif ((score >=0) and (score<60)):
      grade = "E"
else:
      grade = "Wrong"
print("你的考试等级为:{}".format(grade))
```

4.条件判断及组合

表 1 用于条件组合的三个保留字

操作符及使用	描述
x and y	只有当 x，y 均为 True 时，才为 True
x or y	x,y 中有一个为 True，就为 True
not x	非 x

5.程序的异常处理

格式：

try:

<语句块 1>

except:

<语句块 2>

上面这种类型是出现异常都执行后面的。下面这种格式是只有出现指定异常类型才执行这种操作：

try:

<语句块 1>

except <异常类型>:

<语句块 2>

异常处理的高级使用：

try:

<yujukuai1>

except:

<yujukuai2>

else:#不发生异常时执行

<yujukuai3>

finally:#一定执行

<yujukuai4>

4.2 实例 5：身体质量指数 BMI

1.问题需求分析

BMI = 体重（kg）/身高（m）^{2}

表 2 BMI 标准

分类	国际 BMI 值	国内 BMI 值
偏瘦	<18.5	<18.5
正常	18.5~25	18.5~24
偏胖	25~30	24~28
肥胖	≥30	≥28

2.代码：

```
mess = eval(input("请输入你的体重（单位：千克）：\n"))
length = eval(input("请输入你的身高（单位：米）：\n"))
bmi = mess/(length**2)
print("BMI 数值为:{:.2f}".format(bmi))
hcn,hin = "",""
if bmi <18.5:
    hcn = "偏瘦"
    hin = "偏瘦"
elif bmi>=18.5 and bmi <24:
    hcn = "正常"
```

```
    hin = "正常"
elif bmi>=24 and bmi<25:
    hcn,hin = "偏胖","正常"
elif bmi >=25 and bmi<28:
    hcn,hin = "偏胖","偏胖"
elif bmi>=28 and bmi<30:
    hcn, hin ="肥胖","偏胖"
else:
    hcn,hin = "肥胖","肥胖"
print("根据国内 BMI 标准，你的体质为{0}；根据国际 BMI 标准，你的体质为{1}。
".format(hcn,hin))
```

4.3 程序的循环结构

1.遍历循环

常用使用方式：

（1）计数循环（N 次）：

for i in range(N):

<语句块>

for I in range(M,N,K):

<语句块>

这里面，M 代表起始数字，N 代表生成到 N-1，K 为步长。

（2）字符串遍历应用

for c in s:

<语句块>

s 是字符串，遍历字符串中每个字符，产生循环。

（3）列表遍历循环

for item in ls:

<语句块>

其中 ls 是一个列表，遍历其每一个元素，产生循环。

（4）文件遍历循环

for line in fi:

<语句块>

fi 是一个文件标识符，遍历其每行，产生循环。

2.无限循环

while<条件>:

<语句块>

注意：若程序循环执行不退出，则可以按 Ctrl+C 退出程序。

3.循环控制保留字

break👉跳出并结束当前**整个**循环，执行循环后面的语句；

continue👉跳出并结束**本次**循环，执行下一次循环。

4.循环的高级用法

循环与 else

当循环没有被 break 语句退出时，执行 else 语句块。else 语句块作为“正常”完成循环的

奖励，这里面的 else 用法和异常中类似。

例如：

```
for c in "PYTHON":
    if c =="T":
        continue
    print(c,end="")
else:
    print("正常退出")
```

运行结果：PYHON 正常退出

4.4 random 库的使用

1.random 库基本介绍

random 库是使用随机数的 python 标准库。

基本随机数函数：seed()/random()

扩展随机数函数：randint()/getandbits()/uniform()/randrange()/choice()/shuffle()

2.基本随机数函数

计算机生成随机数，是用随机数种子，根据梅森旋转算法产生。随机数种子确定了随机数的产生。

表 3 基本随机数函数

函数	描述
seed(a=None)	初始化给定的随机数种子，默认使用当前系统时间。如：random.seed(10)是产生种子 10 对应的序列
random()	生成[0.0,1.0]之间的随机小数

3.拓展随机数函数

表 4 扩展随机数函数

函数	描述
randint(a,b)	随机生成一个[a,b]之间的整数
randange(m,n[,k])	生成一个[m,n]之间以 k 为步长的随机整数
getandbits(k)	生成一个 k 比特长的随机整数
uniform(a,b)	生成一个[a,b]之间的随机小数
choice(seq)	从序列 seq 中随机选择一个元素
shuffle(seq)	将序列 seq 中的元素随机排序，返回打乱后的序列

4.5 实例 6：圆周率的计算

1.公式法：

```
pi = 0
N = 100
```

```
for k in range(N):
    pi += 1/pow(16,k)*(\
                4/(8*k+1)-2/(8*k+4)-\
                1/(8*k+5)-1/(8*k+6))
    print("圆周率值是: {}".format(pi))
```

2.蒙特卡洛法