《Python 程序设计(第 3 版)》

程序源代码

黄锐军编著

第 1 章 Python 程序基础

1.1 Python 程序开发环境

1.2 Python 程序语句

例 1-2-1： 简单的 Python 程序。

```
print("Hi,everyone")
print("My first application")
print("Is it simple?")
```

例 1-2-2：输入一个数，计算它的平方根。

```
mport math
s=input("输入一个数:")
s=float(s)
if s>=0:
    s=math.sqrt(s)
    print("平方根是:",s)
else:
    print("负数不能开平方")
print("The End")
```

例 1-2-3：注释语句的程序

```
'''
这是一个计算平方根的程序
input --- 输入函数
if else --- 判断语句
print --- 输出语句
'''
import math
#input 输入字符串
s=input("输入一个数:")
#s 字符串变成浮点数
s=float(s)
#判断 s
if s>=0:
    s=math.sqrt(s)   #计算平方根
    print("平方根是:",s)
else:
    print("负数不能开平方")
```

```
print("The End")    #程序结束
```

1.3 Python 数据类型

例 1-3-1： 整数的格式化输出。

```
m=12
print("|%d|"    % m)
print("|%4d|"    % m)
print("|%-4d|"    % m)
print("|%04d|"    % m)
print("|%-04d|"    % m)
m=12345
print("|%d|"    % m)
print("|%4d|"    % m)
print("|%-4d|"    % m)
print("|%04d|"    % m)
print("|%-04d|"    % m)
m=1.53
print("|%d|"%m)
```

例 1-3-2：输出日期时间

```
year=2015
month=2
day=1
hour=8
minute=12
second=0
print("Time: %04d-%02d-%02d %02d:%02d:%02d" %
(year,month,day,hour,minute,second))
```

例 1-3-3：输出实数

```
m=12.57432
print("|%f|"    % m)
print("|%8.1f|"    % m)
print("|%8.2f|"    % m)
print("|%-8.1f|"    % m)
print("|%-8.0f|"    % m)
```

例 1-3-4：输出字符串

```
m="ab"
```

```
print("|%s|"    % m)
print("|%8s|"    % m)
print("|%-8s|"    % m)
m="我们"
print("|%s|"    % m)
print("|%8s|"    % m)
print("|%-8s|"    % m)
```

1.4 Python 表达式

例 1-4-1：逻辑运算

```
a=1
b=3
c=2
print(a+b>c)
print(a+c>b)
print(b+c>a)
print(a+b>c and a+c>b and b+c>a)
print(a>b or a>c)
```

1.5 实践项目：学生成绩计算

```
#计算学生成绩
math=input("数学成绩:")
chinese=input("语文成绩:")
english=input("英语成绩:")
math=float(math)
chinese=float(chinese)
english=float(english)
sum=math+chinese+english
print("总分:",sum,"平均:",sum/3)
```

第 2 章 Python 程序语句

例 2-1-1：输入一个整数，判断它是奇数还是偶数。

设输入的整数是 n ，n%2==0 则是偶数，不然为奇数，程序如下：

```
n=input("Enter:")
n=int(n)
if n%2==0:
    print("偶数")
else:
    print("奇数")
```

例 2-1-2：输入一个整数，输出其绝对值。

```
n=input("Enter:")
n=int(n)
if n>=0:
    print(n)
else:
    print(-n)
```

或者：

```
n=input("Enter:")
n=int(n)
print(n) if n>=0 else print(-n)
```

2.2 复杂条件语句

例 2-2-1：输入一个学生的整数成绩 m，按[90,100]、[80，89]、[70，79]、[60，69]、[0，59]的范围分别给出 A、B、C、D、E 的等级。

```
m=input("Enter mark:")
m=float(m)
if m<0 or m>100:
    print("Invalid")
elif m>=90:
    print("A")
elif m>=80:
    print("B")
elif m>=70:
    print("C")
elif m>=60:
    print("D")
else:
    print("E")
```

例 2-2-2：输入 0-6 的整数，把它作为星期，其中 0 对应星期日，1 对应星期一，……，输出 Sunday,Monday,Tuesday,Wednesday,Thursday,Friday,Saturday。

```
w=input("w=")
w=int(w)
if w==0:
    s="Sunday"
elif w==1:
    s="Monday"
elif w==2:
    s="Tuesday"
elif w==3:
    s="Wednesday"
elif w==4:
    s="Thursday"
elif w==5:
    s="Friday"
elif w==6:
    s="Sturday"
else:
    s="Unknown"
print(s)
```

例 2-2-3：输入一元二次方程的系数 a、b、c，求它的根。

```
a=input("a=")
b=input("b=")
c=input("c=")
a=float(a)
b=float(b)
c=float(c)
if a!=0:
    d=b*b-4*a*c
    if d>0:
        d=math.sqrt(d)
        x1=(-b+d)/2/a
        x2 = (-b - d) / 2 / a
        print("x1=",x1,"x2=",x2)
    elif d==0:
        print("x1,x2=", -b/2/a)
    else:
        print("无实数解")
else:
    print("不是一元二次方程")
```

例 2-2-4：学校的作息时间如表 2-2-1 所示，输入一个 24 小时范围的时间，输出目前的作息项目。

```
t=input("请输入时间(0-24):")
t=int(t)
if t<0 or t>24:
    print("无效时间")
elif t>=23 or t<=7:
    print("睡眠休息")
elif t<=8:
    print("早餐")
elif t<=12:
    print("上午课")
elif t<=14:
    print("午餐、午休")
elif t<=17:
    print("下午课")
elif t<=18:
    print("课外活动")
elif t<=19:
    print("晚餐")
elif t<=22:
    print("晚自习")
else:
    print("休息")
```

2.2.3 【案例】月份的最大日期

```
m=input("输入月份(1-12):")
m=int(m)
if m<0 or m>12:
    print("无效月份")
else:
    maxDays=0
    if m==2:
        y=input("输入年份(1900-):")
        y=int(y)
        if y>=1900:
            if y%400==0 or y%4==0 and y%100!=0:
                maxDays=29
            else:
                maxDays=28
        else:
            print("无效年份")
    elif m==4 or m==6 or m==9 or m==11:
        maxDays=30
```

```
        else:
            maxDays=31
        if maxDays!=0:
            print("月份",m,"最大日期",maxDays)
```

2.3 while 循环语句

例 2-3-1：有限次数的循环

```
n=0
while n<3:
    print(n)
    n=n+1
print(“Last”,n)
```

例 2-3-2：死循环

```
i=0
while i<4:
    print(i)
```

例 2-3-3：计算 s=1+2+3+.....+n 的和，其中 n 由键盘输入。

```
n=input()
n=int(n)
s=0
m=1
while m<=n:
    s=s+m
    m=m+1
print(s)
```

例 2-3-4：输入 5 个同学的成绩，计算平均成绩。

```
s=0
i=0
while i<5:
    m=input("第"+str(i+1)+"个成绩: ")
    m=float(m)
    s=s+m
    i=i+1
print("平均成绩:",s/5)
```

例 2-3-5：输入一个正整数，按相反的数字顺序输出另一个数，例如输入 3621，输出 1263。

```
n=input("n=")
```

```
n=int(n)
s=""
while n!=0:
    m=n%10
    s=s+str(m)
    n=n//10
print(s)
```

2.3.3 【案例】有理数除法的精确计算

```
#输入 p
p=0
while p<=0:
    p=input("Enter p:")
    p=int(p)
#输入 q
q=0
while q<=0:
    q=input("Enter q:")
    q=int(q)
#输入 n
n=0
while n<=0:
    n=input("Enter n:")
    n=int(n)
#p/q
s=str(p//q)
r=p%q
if r!=0:
    s=s+"."
i=0
while r!=0 and i<n:
    r=10*r
    s=s+str(r//q)
    r=r%q
    i=i+1
print("%d/%d=%f" % (p,q,p/q))
print("%d/%d=%s" %(p,q,s))
```

2.4 while 循环的退出

例 2-4-1：输入一个正整数，判断它是否为一个素数（质数）。

```
n=input("n=")
n=int(n)
m=2
while m<n:
    if n%m==0:
        break
    m=m+1
if m==n:
    print(n," 是素数")
else:
    print(n," 不是素数")
```

例 2-4-2：输入两个正整数，找出它们的最小公倍数。

```
a=input("a=")
b=input("b=")
a=int(a)
b=int(b)
if a>b:
    c=a
else:
    c=b
m=a*b
while c<=m:
    if c%a==0 and c%b==0:
        break
    c=c+1
print(c)
```

例 2-4-3：输入两个正整数，找出它们的最大公约数。

```
a=input("a=")
b=input("b=")
a=int(a)
b=int(b)
if a>b:
    c=b
else:
    c=a
while c>=1:
    if a%c==0 and b%c==0:
        break
```

```
        c=c-1
print(c)
```

例 2-4-4：计算圆周率的近似值。

```
import math
m=input("请输入计算次数:")
m=int(m)
s=0
n=1
while n<=m:
        s=s+1/n/n
        n=n+1
pi=math.sqrt(6*s)
print("圆周率=",pi)
```

2.4.3 【案例】输入学生成绩

```
while True:
        m=input("Enter mark [0,100]:")
        m=float(m)
        if m>=0 and m<=100:
                break
print("mark=",m)
```

2.5 for 循环语句

2.5.4 【案例】计算数值和

```
#输入 a
a=0
while a<=0 or a>=10:
        a=input("Enter a[1,9]:")
        a=int(a)
#输入 a
n=0
while n<=0:
        n=input("Enter n:")
        n=int(n)
m=0
s=0
for i in range(n):
        m=10*m+a
```

```
        s=s+m
        if i<n-1:
            print(m,end="+")
        else:
            print(m,end="=")
print(s)
```

2.6 循环注意事项

2.6.4 【案例】数的进制转换

```
while True:
    n=input("输入一个正整数：")
    n=int(n)
    if n<=0:
        print("输入错误，程序退出！")
        break
    print("十进制：",n)
    s=""
    while n>0:
        r=n%16
        if r==10:
            s="A"+s
        elif r==11:
            s="B"+s
        elif r==12:
            s="C"+s
        elif r==13:
            s="D"+s
        elif r==14:
            s="E"+s
        elif r==15:
            s="F"+s
        else:
            s=str(r)+s
        n=n//16
    print("十六进制：",s)
```

2.7 循环的嵌套

例 2-7-1：打印九九乘法表。

```
for i in range(1,10):
    for j in range(1,i+1):
        print("%d*%d=%-4d"%(i,j,i*j),end="")
    print()
```

例 2-7-2：找出 2~100 之间的所有素数。

```
count=0
for n in range(2,101):
    #flag 标志素数
    flag=1
    for m in range(2,n):
        if n%m==0:
            #如果能除尽，那么 n 不是素数，flag=0,退出 m 的内循环
            flag=0
            break
    if flag==1:
        print("%5d" % n,end="")
        count+=1
        if count%5==0:
            print()
```

例 2-7-3：打印出 1,2,3 的这样 3 个数字的所有排列。

```
for i in range(1,4):
   for j in range(1,4):
      for k in range(1,4):
         if i!=j and j!=k and i!=k:
            print(i,j,k)
```

例 2-7-4：中国古代数学家张丘建在他的《算经》中提出了著名的“百钱买百鸡问题”：鸡翁一，值钱五，鸡母一，值钱三，鸡雏三，值钱一，百钱买百鸡，问翁、母、雏各几何？

```
for a in range(21):
    for b in range(34):
        c=100-a-b
        if c>=0 and c%3==0 and 5*a+3*b+c//3==100:
            print("鸡翁%d, 鸡母%d, 鸡雏%d" % (a,b,c))
```

例 2-7-5：数学上一个有趣的现象：任给一个自然数，若为偶数除以 2，若为奇数则乘 3 加 1，得到一个新的自然数后按照上面的法则继续演算，若干次后得到的结果必然为 1，请编程验证。

```
n=int(input("Enter a number: "))
```

```
while n!=1:
    if n%2==0:
        n=n//2
    else:
        n=3*n+1
    print(n, end=" ")
print(n)
```

例 2-7-6：若一个口袋中放有 7 个球，其中有 2 个红的，2 个白的和 3 个黑的，问从中任取 4 个共有多少种不同的颜色搭配？

```
for r in range(3):
    for w in range(3):
        b=4-r-w
        if b>=0 and b<=3:
            print("(r,w,b)-(%d,%d,%d)"%(r,w,b),end=" ")
```

例 2-7-7：有两个乒乓球队伍的队员进行捉对比赛，第一个队伍有[a,b,c,d,e]共 5 个队员，第二个队伍有[1,2,3,4,5]编号的 5 个队员，现在有甲乙丙丁 4 位工作人员进行谈话如下：

甲说：b 与 5 比，d 与 3 比；

乙说：a 与 1 比，e 与 4 比；

丙说：c 与 4 比，d 与 2 比；

丁说：b 与 3 比，e 与 5 比；

核实后结果发现每个人说对了一半，请问两个队伍的捉对比赛是怎么样安排的？

```
for a in range(1,6):
    for b in range(1,6):
        for c in range(1,6):
            for d in range(1,6):
                for e in range(1,6):
                    if a!=b and a!=c and a!=d and a!=e and b!=c and b!=d and b!=e and c!=d and c!=e and d!=e:
                        c1=(b==5 or d==3 ) and not (b==5 and d==3)
                        c2=(a==1 or e==4 ) and not (a==1 and e==4)
                        c3=(c==4 or d==2 ) and not (c==4 and d==2)
                        c4=(b==3 or e==5 ) and not (b==3 and e==5)
                        if c1 and c2 and c3 and c4:
                            print("比赛安排如下：")
                            print("a vs",a)
                            print("b vs",b)
                            print("c vs",c)
                            print("d vs",d)
                            print("e vs",e)
```

例 2-7-8：打印以下图案

```
*
***
```

*****
****

```
for i in range(4):
    for j in range(2*i+1):
        print("*",end="")
    print()
```

例 2-7-9：设 N 是一个四位数，它的 9 倍恰好是其反序数，求 N。

```
for n in range(1000,1112):
        a=n%10
        b=(n//10)%10
        c=(n//100)%10
        d=n//1000
        m=a*1000+b*100+c*10+d
        if n*9==m:
                print(n,"->",m)
                break
```

2.7.5 【案例】整数的素因素分解

```
n=input("Enter an integer:")
n=int(n)
first=True
print(n,end="")
i=2
while i<=n:
        while n%i==0:
                if first:
                        print("=",i,end="")
                        first=False
                else:
                        print("*",i,end="")
                n=n//i
        i=i+1
```

2.8 异常处理

例 2-8-1：输入一个数，计算它的平方根。

```
import math
n=input("Enter:")
```

```
n=float(n)
print(math.sqrt(n))
print("done")
```

例 2-8-2：异常语句

```
print("start")
try:
    print("divided")
    n=1/0
    print("finish")
except Exception as err:
    print(err)
print("end")
```

例 2-8-3：用 raise 语句抛出异常

```
print("start")
try:
    print("In try")
    raise Exception("My error")
    print("finish")
except Exception as err:
    print(err)
print("end")
```

例 2-8-4：应用异常处理，输入一个整数，计算它的平方根。

```
import math
while True:
    try:
        n=input("Enter: ")
        n=int(n)
        if n<0:
            raise Exception("整数为负数")
        print(math.sqrt(n))
        print("done")
        break
    except Exception as err:
        print("输入错误: ",err)
```

例 2-8-5：应用异常处理，输入一个整数，计算它的平方根。

```
import math
while True:
    try:
        n=input("Enter: ")
        n=int(n)
        if n<0:
            raise Exception()
        break
    except:
```

```
                print("请输入正整数")
print(math.sqrt(n))
print("done")
```

例 2-8-6：输入学生信息。

```
try:
    Name=input("姓名:")
    if Name.strip()=="":
        raise Exception("无效的姓名")
    Gender=input("性别:")
    if Gender!="男" and Gender!="女":
        raise Exception("无效的性别")
    Age = input("年龄:")
    Age=float(Age)
    if Age<18 or Age>30:
        raise Exception("无效的年龄")
    print(Name,Gender,Age)
except Exception as err:
    print(err)
```

2.8.6 【案例】学生成绩计算

```
#计算学生成绩
math=input("数学成绩:")
chinese=input("语文成绩:")
english=input("英语成绩:")
try:
    math=float(math)
    chinese=float(chinese)
    english=float(english)
    if math<0 or math>100 or chinese<0 or chinese>100 or english<0 or english>100:
        raise Exception("成绩输入有误！")
    sum=math+chinese+english
    print("总分:",sum,"平均:",sum/3)
except Exception as err:
    print(err)
```

2.9 实践项目 验证哥德巴赫猜想

```
#输入偶数，如果不满足要求就继续输入
```

```
while True:
    n=input("输入 6 以上的偶数:")
    n=int(n)
    if n%2==0 and n>=6:
        break
#p 的最大值 maxp
maxp=n//2
p=3
while p<=maxp:
    #判断 p 是否是素数，flag=True 表示是素数
    flag=True
    for k in range(2,p):
        if p%k==0:
            #p 可以被比它小的整数除尽，不是素数
            flag=False
            break
    #如果 p 是素数，再次判断 q 是否时素数
    if flag:
        q=n-p
        for k in range(2,q):
            if q %k==0:
                # q 可以被比它小的整数除尽，不是素数
                flag=False
                break
        #q 也是素数，得到一个分解 n=p+q
        if flag:
            print(n,"=",p,"+",q)
    p=p+1
print("done!")
```

第 3 章 Python 函数与模块

3.1 Python 函数

例 3-1-1：输入 2 个整数，找出它们中的最大值。

```
def max(a,b):
    c=a
    if b>a:
        c=b
    return c
#调用 max
m=max(2,4)
print(m)
```

例 3-1-2：输入 3 个整数，找出它们中的最大值。

```
def max(a,b):
    c=a
    if b>a:
        b
    return c

a=input("a=")
b=input("b=")
c=input("c=")
a=float(a)
b=float(b)
c=float(c)
d=max(a,b)
e=max(d,c)
print("max=",e)
```

例 3-1-3： 两个乒乓球队进行比赛，各出三人。甲队为 a,b,c 三人，乙队为 x,y,z 三人。已抽签决定比赛名单。有人向队员打听比赛的名单。a 说他不和 x 比，c 说他不和 x,z 比，请编程序找出三队赛手的名单。

```
def xyz(v):
    if v==0:
        return "x"
    elif v==1:
        return "y"
    else:
        return "z"

for a in range(3):
```

```
    for b in range(3):
        for c in range(3):
            if a!=b and a!=c and b!=c and a!=0 and c!=0 and c!=2:
                print("a vs",xyz(a))
                print("b vs",xyz(b))
                print("c vs",xyz(c))
```

例 3-1-4：如果整数 A 的全部因子(包括 1，不包括 A 本身)之和等于 B；且整数 B 的全部因子(包括 1，不包括 B 本身)之和等于 A，则将整数 A 和 B 称为亲密数。编程找出 2000 内的全部亲密数对。

```
def factorSum(m):
    s=0
    for i in range(1,m):
        if m%i==0:
            s+=i
    return s

for a in range(1,2001):
    for b in range(a+1,2001):
        if factorSum(a)==b and factorSum(b)==a:
            print("亲密数对  (%d,%d)"%(a,b))
```

3.1.4 【案例】 最大公约数与最小公倍数

```
#最大公约数函数
#d 最小为 1，必定会返回
def maxDivider(a,b):
    c=a
    if b<a:
        c=b
    for d in range(c,0,-1):
        if a%d==0 and b%d==0:
            return d

#最小公倍数函数
#d 最多为 a*b，必定返回
def minMultiplier(a,b):
    c=a
    if b>a:
        c=b
    m=a*b
    for d in range(c,m+1,1):
        if d%a==0 and d%b==0:
            return d
```

```
#主程序
a=input("a=")
b=input("b=")
a=int(a)
b=int(b)
print("最大公约数",maxDivider(a,b))
print("最小公倍数",minMultiplier(a,b))
```

3.2 Python 变量范围

例 3-2-1：局部变量。

```
def sum(m):
        s=0
        for p in range(m+1):
                s=s+p
                #  计算(1+2+……m)的和
 return s

m=10
s=sum(m)
print(s)
```

例 3-2-2：局部变量。

```
def fun(x, y) :
        print("In fun:",x,y)
         x=1
         y=2

x=100
y=200
fun(x,y)
print(x,y)
```

例 3-2-3：全局变量。

```
def fun(x):
        global y
        y=0
        x=0

x=1
y=2
fun(x)
print(x,y)
```

例 3-2-4：全局变量。

```
def A(x):
    global y
    y=0
    x=0

def B(x):
    global y
    y=10
    x=0

x=1
y=2
A(x)
B(x)
print(x,y)
```

例 3-2-5：全局变量。

```
def fun(x):
    print(x,y)

y=1
fun(2)
```

例 3-2-6：全局变量。

```
def fun(x):
    print(x,y)
    y=100

y=1
fun(2)
```

3.2.4 【案例】银行存款与取款过程。

```
ef deposit():
    global balance
    amount=input("存款金额:")
    amount=float(amount)
    balance += amount

def withdraw():
    global balance
    amount=input("取款金额:")
    amount=float(amount)
    if balance >= amount:
```

```
            balance -= amount
        else:
            print("余额不足")

balance = 0
while True:
    print("1.存入")
    print("2.支取")
    print("3.退出")
    print("您的余额为:", balance)
    choice = input("请选择:")
    if choice == "1":
        deposit()
    elif choice == "2":
        withdraw()
    elif choice == "3":
        break
```

3.3 函数调用

例 3-3-1：输入整数 n，计算 1+(1+2)+(1+2+3)+……+(1+2+3+……+n)的和。

```
def sum(m):
    s=0
    for n in range(1,m+1):
        s=s+n
    return s

def sumAll(n):
    s=0
    for m in range(1,n+1):
        s=s+sum(m)
    return s

n=input("n=")
n=int(n)
print("总和是",sumAll(n))
```

例 3-3-2：输入一个正整数,找出它的所有素数因素。

```
def IsPrime(m):
    for n in range(2,m):
        if m%n==0:
            return 0
```

```
        return 1

n=input("n=")
n=int(n)
for p in range(2,n+1):
        if n%p==0 and IsPrime(p)==1:
                print(p)
```

例 3-3-3：输入一个正整数 n,计算 s=1+2+3+...+n 的值。

```
def S(n):
        if n==1:
                return 1
        else:
                return S(n-1)+n

n=input("n=")
n=int(n)
print("S(%d)=%d" %(n,S(n)))
```

3.3.4 【案例】验证歌德巴赫猜想

```
def IsPrime(m):
        for n in range(2,m):
                if m%n==0:
                        return 0
        return 1

for n in range(6,101,2):
        for p in range(3,n+1,2):
                q=n-p
                if IsPrime(p) and IsPrime(q):
                        print(n,"=",p,"+",q)
                        break
```

3.4 函数默认参数

例 3-4-1：默认参数的函数

```
def fun(a,b=1,c=2):
        print(a,b,c)

fun(0)
fun(1,2)
fun(1,2,3)
```

例 3-4-2：参数按名称指定

```
def fun(a,b=1,c=2):
    print(a,b,c)

fun(0,c=4,b=2)
fun(0,c=4)
fun(b=2,a=1,c=4)
fun(a=0,c=4,b=2)
fun(c=1,b=3,a=2)
```

3.4.4 【案例】print 函数的默认参数

```
print(1,2)
print(1,2,sep='-')
print("line")
print('line',end='*')
print('end')
```

3.5 函数与异常

例 3-5-1：函数的异常扑捉

```
def fun():
    print("start")
    n=1/0
    print("end")

try:
    fun()
except Exception as err:
    print(err)
```

例 3-5-2：函数的异常扑捉

```
def fun():
    print("start")
    try:
        n=1/0
        print("end")
    except:
        print("error")

try:
    fun()
```

```
except Exception as err:
    print(err)
```

例 3-5-3：异常的传递

```
def A():
    print("start A")
    n=1/0
    print("end A")

def B():
    print("start B")
    A()
    print("end B")
try:
    B()
    print("done")
except Exception as err:
    print(err)
print("finish")
```

例 3-5-4：异常的传递

```
def A():
    print("start A")
    n=1/0
    print("end A")

def B():
    print("start B")
    try:
        A()
    except Exception as err:
        print(err)
    print("end B")
try:
    B()
    print("done")
except Exception as err:
    print(err)
print("finish")
```

例 3-5-5：异常终止程序

```
def A():
```

```
        print("start A")
        n=1/0
        print("end A")

def B():
        print("start B")
        A()
        print("end B")

B()
print("finish")
```

3.5.3 【案例】时间的输入与显示

```
def myTime():
    h=input("时:")
    h=int(h)
    if h<0 or h>23:
        raise Exception("无效的时")
    m=input("分:")
    m=int(m)
    if m<0 or m>59:
        raise Exception("无效的分")
    s=input("秒:")
    s=int(s)
    if s<0 or s>59:
        raise Exception("无效的秒")
    print("%02d:%02d:%02d" %(h,m,s))

try:
    myTime()
except Exception as e:
    print(e)
```

3.6 Python 模块

例 3-6-1：设计模块并引用它

```
def myMin(a,b):
    c=a
    if a>b:
        c=b
```

```
        return c

def myMax(a,b):
        c=a
        if a<b:
                c=b
        return c
```

例 3-6-2：设计模块放在子目录中并引用它

```
from mine.myModule import myMin,myMax
print(myMin(1,2),myMax(1,2))
```

例 3-6-3：设计模块放在 lib 目录中并引用它

```
from myModule import myMin,myMax
print(myMin(1,2),myMax(1,2))
```

3.6.4 【案例】测试 Python 模块的位置

```
import sys
paths=sys.path
for p in paths:
        print(p)
```

3.7 实践项目 打印万年日历

```
#打印日历
def isLeap(y):
        return y%400==0 or y%4==0 and y%100!=0

def maxDays(y,m):
        d=30
        if m==1 or m==3 or m==5 or m==7 or m==8 or m==10 or m==12:
                d=31
        elif m==2:
                d=29 if isLeap(y) else 28
        return d

def countWeek(y):
        w=(y-1)+(y-1)//400+(y-1)//4-(y-1)//100+1
        return w%7

def printMonth(w,y,m):
        md=maxDays(y,m)
        print("%-6s%-6s%-6s%-6s%-6s%-6s%-6s%-
```

```
6s" %("Sun","Mon","Tue","Wed","Thu","Fri","Sat"))
        for i in range(w):
            print("%-6s" % " ",end="")
        for d in range(1,md+1):
            print("%-6d" % d,end="")
            w=w+1
            if w%7==0:
                print()
        return w%7

    y=input("输入年份:")
    y=int(y)
    w=countWeek(y)
    for m in range(1,13):
        print()
        print("-------------",y,"年",m,"月  -------------")
        w=printMonth(w,y,m)
        print()
```

第 4 章 Python 序列数据

4.1 字符串类型

例 4-1-1：输入一个字符串，统计它包含的大写字母的个数

```
s=input("Enter a string: ")
count=0
for i in range(len(s)):
    if s[i]>="A" and s[i]<="Z":
        count=count+1
print("count=",count)
```

例 4-1-2：输入一个字符串，统计它包含的大写字母、小写字母、数字的个数

```
s=input("Enter a string: ")
upper=0
lower=0
digit=0
for i in range(len(s)):
    if s[i]>="A" and s[i]<="Z":
        upper=upper+1
    elif s[i]>="a" and s[i]<="z":
        lower=lower+1
    elif s[i]>="0" and s[i]<="9":
        digit=digit+1
print("Upper chars: ",upper)
print("Lower chars: ",lower)
print("Digit chars: ",digit)
```

例 4-1-3：输入一个字符串，把它反向显示

```
def reverseA(s):
    t=""
    for i in range(len(s)-1,-1,-1):
        t=t+s[i]
    return t

def reverseB(s):
    t=""
    for i in range(0,len(s)):
        t=s[i]+t
    return t
```

```
print(reverseA("reverse"))
print(reverseB("reverse"))
```

例 4-1-4：输入一个字符串，去掉它左右多余的空格，例如" a bc "返回"a bc"。

```
def trim(s):
    t=""
    i=0
    j=len(s)-1
    while i<=j and s[i]==" ":
        i=i+1
    while i<=j and s[j]==" ":
        j=j-1
    for k in range(i,j+1):
        t=t+s[k]
    return t

s=input("Enter a string: ")
print(s,"length=",len(s))
t=trim(s)
print(t,"length=",len(t))
```

例 4-1-5：编写一个把字符串中所有小写字母变大写的函数和另一个大写字母变小写的函数

```
def myToUpper(s):
    t=""
    for i in range(len(s)):
        if s[i]>="a" and s[i]<="z":
            t=t+chr(ord("A")+ord(s[i])-ord("a"))
        else:
            t=t+s[i]
    return t

def myToLower(s):
    t=""
    for i in range(len(s)):
        if s[i]>="A" and s[i]<="Z":
            t=t+chr(ord("a")+ord(s[i])-ord("A"))
        else:
            t=t+s[i]
    return t

s=input("Enter a string: ")
print("myToUpper: ",myToUpper(s))
```

```
print("myToLower: ",myToLower(s))
```

例 4-1-6： 设计函数 subString(s,start,length)表示从 s 的 start 位置开始，取出长度为 length 的一个子串。

```
def subString(s,start,length):
    m=len(s)
    if start>=length:
        return ""
    t=""
    i=start
    while i<start+length and i<m:
        t=t+s[i]
        i=i+1
    return t

s="abcdefghijk"
print(subString(s,2,4))
print(subString(s,2,24))
```

4.1.3 【案例】字符串的对称

```
def reverse(s):
    t=""
    for i in range(len(s)-1,-1,-1):
            t=t+s[i]
    return t

def isSymmetry(s):
    t=reverse(s)
    if s==t:
            return 1
    else:
            return 0

s=input("Enter a string: ")
if isSymmetry(s)==1:
        print("对称")
else:
        print("不对称")
```

4.2 字符串函数

例 4-2-1：编写 myLower(s)实现 s.lower()

```
def myLower(s):
    t=""
    for i in range(len(s)):
        if s[i]>="A" and s[i]<="Z":
            t=t+chr(ord("a")+ord(s[i])-ord("A"))
        else:
            t=t+s[i]
    return t

s="aABEbwWFEW"
a=s.lower()
b=myLower(s)
print(a)
print(b)
```

例 4-2-2：编写 myStrip(s)实现 s.strip()

```
def myStrip(s):
    i=0
    j=len(s)-1
    #i 是左边下标，跳过左边空格
    while i<=j and s[i]==" ":
        i=i+1
    #j 是右边下标，跳过右边空格
    while j>=i and s[j]==" ":
        j=j-1
    return s[i:j+1]

s=" a b   "
a=myStrip(s)
b=s.strip()
print(a,len(a))
print(b,len(b))
```

例 4-2-3：编写 mySplit(s,sep)实现 s.split(sep)

```
def mySplit(s,sep):
    i=s.find(sep)
    t=[]
    while i>=0:
        w=s[0:i]
        t.append(w)
        s=s[i+len(sep):]
        i=s.find(sep)
    t.append(s)
    return t
```

```
s="abcababcab"
a=s.split("ab")
b=mySplit(s,"ab")
print(a)
print(b)
```

4.2.3 【案例】寻找字符串的子串

```
def myFind(s,t):
    m=len(s)
    n=len(t)
    if m<n:
        return -1
    i=0
    while i<=m-n:
        j=0
        while j<n:
            if s[i+j]!=t[j]:
                break
            j=j+1
        if j==n:
            return i
        i=i+1
    return -1

s="ababcabcd12ab"
print(myFind(s,"abc"),s.find("abc"))
print(myFind(s,"ad"),s.find("ad"))
```

4.3 列表类型

例 4-3-1：列表的截取

```
L =  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
L[0:2]    #[1, 2],    取区间[i,j) ，左闭右开
L[:2]      #同上，可省略第一位
L:[2:]      #[3, 4, 5, 6, 7, 8, 9, 10, 11]
L[2:-1]    #[3, 4, 5, 6, 7, 8, 9, 10]
L[:]        #同 l1，相当于复制一份
L[::2]    #步长 2，[1, 3, 5, 7, 9, 11]
L[0:7:2]   #[1, 3, 5, 7]
```

```
L[7:0:-2]   #[8, 6, 4, 2]    注意步长为负、理解起来相当于从 7 到 1，倒序步长 2
L[8:14]    #[9,10,11]   注意 end 超过最后的索引
```

例 4-3-2：列表的截取

```
L =   ["a0","a1","a2","a3","a4","a5","a6","a7","a8","a9"]
print("L---",L)
print("L[0:-2]---",L[0:-2])
print("L[:-2]---",L[:-2])
print("L[-2:]---",L[-2:])
print("L[-2,6]---",L[-2:6])
print("L[:]---",L[:])
print("L[::,-2]---",L[::-2])
print("L[7,-1:-1]---",L[7:-1:-1])
print("L[8:0.-1])---",L[8:0:-1])
print("L[5:1:-2]---",L[5:1:-2])
print("L[4:1.-2]---",L[4:1:-2])
```

例 4-3-3：快速生成小写英文字母列表

```
lowerList=[chr(i+ord('a')) for i in range(26)]
print("小写字母列表：",lowerList)
```

例 4-3-4：击鼓传花游戏：n 个人围成一圈，顺序排号，从第一个人开始报数（从 1 到 3 报数），凡报到 3 的人退出圈子，问最后留下的是原来第几号的那位。

```
n=input("输入人数: ")
n=int(n)
Q=[i for i in range(1,n+1)]
print(Q)
i=0
while len(Q)!=1:
    print("开始:",Q[i],end=" -> ")
    i=(i+2)%len(Q)
    print("删除:",Q[i])
    Q.pop(i)
    # 循环结束后，i 指向最后一个元素，需要重新指向第一个元素
    if i>=len(Q):
        i=0
    print(Q)
print("最后剩下一个人:",Q)
```

例 4-3-5：三对男女混合双打比赛，三个男队员 A、B、C，三个女队员 X、Y、Z。A 说他将和 X 搭档；X 说她和 C 搭档；C 说他将和 Z 搭档。现在知道这 3 人的话全是假话。请编程找出谁将和谁搭档。

```
xyz=["x","y","z"]
for a in xyz:
    for b in xyz:
```

```
            for c in xyz:
                if a!="x" and c!="z" and a!=b and b!=c and c!=a:
                    print("(a 搭档",a,"),(b 搭档",b,"),(c 搭档",c,")")
```

例 4-3-6：有 m 个外形一模一样的小球编号 1 到 m，放在一个盒子中任意摸一个，记住它的号码后放回盒子中，继续摸下一个球，这样重复 n 次(n<m)，试用计算机模拟这个过程，并估算摸取 n 次中有重复编号的概率。

```
import random

def A(m,n):
    s=1
    for k in range(m-n+1,m+1):
        s*=k
    return s

m=int(input("输入小球的数量:"))
n=int(input("输入摸取的数量:"))
count=int(input("输入模拟次数:"))
eq=0
for i in range(count):
    selected=[]
    for j in range(n):
        k=random.randint(1,m)
        if k in selected:
            eq+=1
            break
        else:
            selected.append(k)
print("模拟计算概率:%.6f"%(eq/count))
print("数学计算概率:%.6f"%(1-A(m,n)/m**n))
```

例 4-3-6： 列表作为函数参数

```
def fun( mylist,m,s):
    "修改传入的列表"
    mylist.append(1)
    mylist[0] = 100
    m=1
    s="changed"
    print ("函数内取值: ", mylist,m,s)

# 调用 fun 函数
mylist = [10,20,30]
m=0
s="try"
```

```
print ("函数外取值: ", mylist,m,s)
fun( mylist,m,s )
print ("函数外取值: ", mylist,m,s)
```

例 4-3-7： 函数返回列表

```
def fun():
        list=[]
        for i in range(10):
                list.append(i)
        return list

# 调用 fun 函数
list=fun()
print(list)
```

4.3.4 【案例】列表存储省份与城市

```
provinces=["广东","四川","贵州"]
cities=[["广州","深圳","惠州","珠海"],["成都","内江","乐山"],["贵阳","六盘水","遵义"]]
p=input("输入省份:")
found=False
for i in range(len(provinces)):
    if provinces[i]==p:
        print(provinces[i],end=":")
        for j in range(len(cities[i])):
            print(cities[i][j],end=" ")
        found=True
        break
if not found:
    print("没有这个省份")
```

4.4 元组类型

例 4-4-1：元组的使用

```
s=(1,3,2,3,4,5)
print(s)
prnt(type(s))
```

例：4-4-2： 建立一个代表星期的元组表，输入一个 0-6 的整数，输出对应的星期名称。

```
week=("日","一","二","三","四","五","六")
print(week)
w=input("Enter an integer: ")
```

```
w=int(w)
if w>=0 and w<=6:
    print("星期"+week[w])
else:
    print("错误输入")
```

例 4-4-3 元组可变参数的函数

```
def fun(x,y,*args):
    print(x,y,"args",args)

fun(1,2)
fun(1,2,3)
fun(1,2,3,4)
```

4.4.3 【案例】通用的最大值函数

```
def max(*args):
    print(args)
    m=args[0]
    for i in range(len(args)):
        if m<args[i]:
            m=args[i]
    return m

print(max(1,2))
print(max(1,2,0,3))
```

4.5 字典类型

4.5.3 【案例】字典存储学生信息

```
st=[]
def getStudents():
    global st
    st=[]
    st.append({"Name":"张三","Gender":"男","Age":20})
    st.append({"Name": "李四", "Gender": "女", "Age": 21})
    st.append({"Name": "王五", "Gender": "男", "Age": 22})

def seekStudent(Name):
    for s in st:
```

```
            if s["Name"]==Name:
                print(s["Name"], s["Gender"], s["Age"])
                return
        print("没有姓名是",Name,"的学生")

getStudents()
seekStudent("张三")
seekStudent("张四")
```

4.6 字典与函数

例 4-6-1：字典作为函数参数

```
def fun(dict):
    dict["name"]="aaa"
    print("inside:",dict)

dict={"name":"xxx","age":30}
print("before",dict)
fun(dict)
print("after",dict)
```

例 4-6-2：字典作为函数返回值

```
def fun():
    dict={}
    dict["name"]="aaa"
    dict["age"]=20
    dict["gender"]="male"
    return dict

def show(dict):
        keys=dict.keys()
        for key in keys:
            print(key,dict[key])

dict=fun()
print(dict)
show(dict)
```

例 4-6-3：具有字典可变参数的函数

```
def fun(x,y=2,**kargs):
    print(x,y,"kargs",kargs)
```

```
fun(1,2)
fun(1,2,z=3)
fun(1,2,a=3,b="demo")
fun(x=1,y=2,z=3)
fun(y=1,x=2,z=5,s="demo")
fun(x=1,z=3)
```

例 4-6-4：具有元组可变参数与字典可变参数的函数

```
def fun(x,y=2,*args,**kargs):
    print(x,y,"args",args,"kargs",kargs)

fun(1,2)
fun(1,2,3,4)
fun(1,2,3,4,z=5,s="demo")
```

4.6.4 【案例】字典存储省份与城市

```
#provinces 是全局的变量
provinces={}

def append(province,cities):
    global provinces
    if province not in provinces.keys():
        provinces[province]=cities
    else:
        print(province+"已经存在")

def show():
    for p in provinces.keys():
        print(p,provinces[p])

def seekProvince(province):
    if province in provinces.keys():
        print(province,end=":")
        for c in provinces[province]:
            print(c,end=" ")
        print()
    else:
        print("没有这个省份")

def seekCity(city):
    for p in provinces.keys():
        if city in provinces[p]:
            print(city+"属于"+p+"省")
```

```
                return
        print("没有这个城市")

append("广东",["广州","深圳"])
append("四川",["成都", "内江", "乐山"])
append("贵州",["贵阳","六盘水","兴义"])
show()
seekProvince("四川")
seekCity("六盘水")
```

4.7 实践项目：我的英文字典

```
#我的英语小字典
words=[{"word":"about","note":"在附近，关于"},{"word":"post","note":"邮寄、投递"}]

def show():
    for w in words:
        print("%-16s : %s" % (w["word"],w["note"]))
    print()

def enter():
    w={}
    w["word"]=input("单词:")
    w["note"]=input("注释:")
    return w

def seek(word):
    i=0
    j=len(words)-1
    while i<=j:
        m=(i+j)//2
        if words[m]["word"] == word:
            print("%-16s : %s" % (word, words[m]["note"]))
            return
        elif words[m]["word"]>word:
            j=m-1
        else:
            i=m+1
    print(word + " ---  查找失败")

def insert(w):
```

```
    global words
    i=0
    j=len(words)-1
    while i<=j:
        m=(i+j)//2
        if words[m]["word"] == w["word"]:
            print(w["word"]+" ---  已经存在")
            return
        elif words[m]["word"]>w["word"]:
            j=m-1
        else:
            i=m+1
    words.insert(i,w)
    print(w["word"] + " ---  增加成功")

def update(w):
    global words
    i=0
    j=len(words)-1
    while i<=j:
        m=(i+j)//2
        if words[m]["word"] == w["word"]:
            words[m]["note"]=w["note"]
            print(w["word"]+" ---  更新成功")
            return
        elif words[m]["word"]>w["word"]:
            j=m-1
        else:
            i=m+1
    print(w["word"] + " ---  查找失败")

def delete(word):
    global words
    i=0
    j=len(words)-1
    while i<=j:
        m=(i+j)//2
        if words[m]["word"] == word:
            del words[m]
            print(word+" ---  删除成功")
            return
        elif words[m]["word"]>word:
            j=m-1
        else:
```

```
                    i=m+1
        print(word + " ---  查找失败")

while True:
        print("1.显示  2.查找  3.增加  4.更新  5.删除  6.退出")
        s=input("请选择(1,2,3,4,5,6):")
        if s=="1":
                show()
        elif s == "2":
                word = input("单词:")
                seek(word)
        elif s=="3":
                w=enter()
                insert(w)
        elif s=="4":
                w=enter()
                update(w)
        elif s=="5":
                word=input("单词:")
                delete(word)
        elif s=="6":
                break
print("Finished")
```

第 5 章 Python 面向对象

5.1 类与对象

例 5-1-1：定义了一个 Person 类并创建对象

```
class Person:
    def __init__(self,n,g,a):
        self.name=n
        self.gender=g
        self.age=a
        print("创建对象:",n,id(self))

    def show(self):
        print("(",self.name,self.gender,self.age,")")

a=Person("A","M",21)
print(id(a))
a.show()
b=Person("B","F",20)
print(id(b))
b.show()
```

例 5-1-2：访问类属性与方法

```
class Person:
    def __init__(self,n,g,a):
        self.name=n
        self.gender=g
        self.age=a

    def show(self):
        print("(",self.name,self.gender,self.age,")")

    def change(self):
        self.age+=1

a=Person("A","M",21)
a.show()
print("(",a.name,a.gender,a.age,")")
a.age+=1
a.show()
```

```
a.change()
a.show()
```

例 5-1-3：类名称限定调用类方法

```
class Person:
    def __init__(self,n,g,a):
        self.name=n
        self.gender=g
        self.age=a

    def show(self):
        print("(",self.name,self.gender,self.age,")")

a=Person("A","M",21)
a.show()
Person.show(a)
```

例 5-1-4：访问权限

```
class Person:
    def __init__(self,n,g,a):
        self.__name=n
        self.gender=g
        self.age=a

    def __str(self):
        return "({},{},{})".format(self.__name,self.gender,self.age)

    def show(self):
        print(self.__str())

a=Person("A","M",21)
a.show()
print(a.gender,a.age)
#Wrong: print(a.__name)
#Wrong: print(a.__str())
```

5.1.5 【案例】图书对象列表

```
class Book:
    def __init__(self,title,author,price):
        self.title=title
        self.author=author
        self.price=price

    def show(self):
```

```
            print(self.title,",",self.author,",",self.price)

books=[]
books.append(Book("Python 程序设计","张三",39.8))
books.append(Book("Java 程序设计","李四",29.3))
books.append(Book("C 程序设计","王五",32))
for book in books:
    book.show()
```

例 5-2-1：实例方法定义

```
class Person:
    def __init__(self,n):
        self.__name = n

    def getName(self):
        return self.__name

    def setName(self,n):
        self.__name=n

p = Person("James")
print(p.getName())
p.setName("Robert")
print(p.getName())
```

例 5-2-2：类方法定义

```
class Person:
    def __init__(self,n):
        self.__name = n

    @classmethod
    def create(cls,n):
        print(cls)
        return cls(n)

    def getName(self):
        return self.__name

p=Person.create("John")
print(p.getName())
p=Person("Jane")
q=p.create("Mary")
print(q.getName())
```

例 5-2-3：静态方法定义

```
class Person:
    def __init__(self,n):
        self.__name = n

    @classmethod
    def create(cls,n):
        return cls(n)

    @staticmethod
    def make(n):
        return Person(n)

    def show(self):
        print(self.__name)

p=Person.create("John")
p.show()
q=Person.make("Mary")
q.show()
```

例 5-2-4：类属性

```
class Person:
    age=23
    def __init__(self,n):
        self.__name = n

    def show(self):
        print(self.__name)

p=Person("John")
q=Person("Mary")
print(Person.age,p.age,q.age)
Person.age=24
print(Person.age,p.age,q.age)
p.age=25
q.age=26
print(Person.age,p.age,q.age)
```

5.2.3 【案例】我的时间类

```
class MyTime:
    def __init__(self,hour,mimute,second):
        if hour>=0 and hour<24 and mimute>=0 and mimute<60 and second>=0 and second<60:
            self.hour=hour
            self.minute=mimute
```

```
                self.second=second
            else:
                raise Exception("无效时间")

        def show(self):
            print("%02d:%02d:%02d"%(self.hour,self.minute,self.second))

        def toString(self):
            return "%02d:%02d:%02d"%(self.hour,self.minute,self.second)

        def toSeconds(self):
            return self.hour*3600+self.minute * 60 + self.second

        @classmethod
        def fromSeconds(cls,seconds):
            h=seconds//3600
            seconds=seconds%3600
            m=seconds//60
            s=seconds%60
            return cls(h,m,s)

        @staticmethod
        def fromString(string):
            h,m,s=string.split(":")
            return MyTime(int(h),int(m),int(s))

        def compare(self,t):
            a=self.toSeconds()
            b=t.toSeconds()
            if a>b:
                return 1
            elif a<b:
                return -1
            else:
                return 0

    a=MyTime.fromSeconds(30000)
    b=MyTime.fromString("10:20:30")
    a.show()
    b.show()
    sa=a.toString()
    sb=b.toString()
    c=a.compare(b)
    if c==0:
```

```
        print(sa," is equal to ",sb)
elif c==-1:
        print(sa,"is ealier than ",sb)
else:
        print(sa," is later than ",sb)
```

5.3 对象初始化

例 5-3-1： 构造方法函数与析构方法函数

```
class Person:
        def __init__(self,n):
                print("__init__",self,n)
                self.name=n

        def __del__(self):
                print("__del__",self)

        def show(self):
                print(self,self.name)

p=Person("james")
p.show()
print(p)
```

例 5-3-2：对象初始化

```
class Person:
        def __init__(self,n,g,a):
                self.name=n
                self.gender=g
                self.age=a

        def show(self):
                print(self.name,self.gender,self.age)

p=Person("james","male",21)
p.show()
```

例 5-3-3：设置__init__中有默认参数

```
class Person:
        def __init__(self,n="",g="male",a=0):
```

```
            self.name=n
            self.gender=g
            self.age=a

        def show(self):
            print(self.name,self.gender,self.age)

a=Person("james")
b=Person("james","female")
c=Person("james","male",20)
a.show()
b.show()
c.show()
```

例 5-3-4：实例方法的 self 参数

```
class Person:
    def __init__(self,n="",g="male",a=0):
        self.name=n
        self.gender=g
        self.age=a

    def show(self):
        print(self)
        print(self.name,self.gender,self.age)

p=Person("james","male",20)
Person.show(p)
p.show()
```

5.3.5【案例】我的日期类

```
class MyDate:
    def __init__(self,y,m,d):
        months=[0,31,28,31,30,31,30,31,31,30,31,30,31]
        if y<0:
            raise Exception("无效年份")
        if m<1 or m>12:
            raise Exception("无效月份")
        if y%400==0 or y%4==0 and y%100!=0:
            months[2]=29
        else:
            months[2]=28
        if d<1 or d>months[m]:
            raise Exception("无效日期")
        self.year=y
```

```
            self.month=m
            self.day=d

        def show(self,end='\n'):
            print("%04d-%02d-%02d" %(self.year,self.month,self.day),end=end)

        @staticmethod
        def fromString(s):
            y,m,d=s.split('-')
            return MyDate(int(y),int(m),int(d))

        def toString(self):
            return "%04d-%02d-%02d" %(self.year,self.month,self.day)

        def compare(self,d):
            if self.year<d.year:
                return -1
            elif self.year>d.year:
                return 1
            else:
                if self.month<d.month:
                    return -1
                elif self.month>d.month:
                    return 1
                else:
                    if self.day<d.day:
                        return -1
                    elif self.day>d.day:
                        return 1
                    else:
                        return 0

    try:
        a=MyDate(2017,7,8)
        b=MyDate.fromString("2024-05-19")
        a.show()
        b.show()
        sa=a.toString()
        sb=b.toString()
        c=a.compare(b)
        if c==0:
            print(sa,"is equal to ",sb)
        elif c==-1:
            print(sa,"is earlier than ",sb)
```

```
        else:
            print(sa,"is later than ",sb)
    except Exception as e:
        print(e)
```

5.4 类的继承

例 5-4-1：派生于继承

```
class Person:
    def __init__(self,name,gender,age):
        self.name=name
        self.gender=gender
        self.age=age

    def show(self,end='\n'):
        print(self.name,self.gender,self.age,end=end)

class Student(Person):
    def __init__(self,name,gender,age,major,dept):
        Person.__init__(self,name,gender,age)
        self.major=major
        self.dept=dept

    def show(self):
        Person.show(self,' ')
        print(self.major,self.dept)

s=Student("james","male",20,"software","computer")
s.show()
```

例 5-4-2：属性方法继承

```
class Person:
    def __init__(self,name,gender,age):
        self.name=name
        self.gender=gender
        self.age=age

    def show(self,end='\n'):
        print(self.name,self.gender,self.age,end=end)

    def display(self,end='\n'):
```

```
            print(self.name,self.gender,self.age,end=end)

class Student(Person):
    def __init__(self,name,gender,age,major,dept):
        Person.__init__(self,name,gender,age)
        self.major=major
        self.dept=dept

    def show(self):
        Person.show(self,' ')
        print(self.major,self.dept)

    def setName(self,name):
        self.name=name

s=Student("james","male",20,"software","computer")
s.show()
s.display()
s.setName("robert")
s.show()
s.display()
```

5.4.3 【案例】我的日期时间类

```
class MyDate:
    def __init__(self,y,m,d):
        months=[0,31,28,31,30,31,30,31,31,30,31,30,31]
        if y<0:
            raise Exception("无效年份")
        if m<1 or m>12:
            raise Exception("无效月份")
        if y%400==0 or y%4==0 and y%100!=0:
            months[2]=29
        else:
            months[2]=28
        if d<1 or d>months[m]:
            raise Exception("无效日期")
        self.year=y
        self.month=m
        self.day=d

    def show(self,end='\n'):
        print("%04d-%02d-%02d" %(self.year,self.month,self.day),end=end)

    @staticmethod
```

```
        def fromString(s):
            y,m,d=s.split('-')
            return MyDate(int(y),int(m),int(d))

        def toString(self):
            return "%04d-%02d-%02d" %(self.year,self.month,self.day)

        def compare(self,d):
            if self.year<d.year:
                return -1
            elif self.year>d.year:
                return 1
            else:
                if self.month<d.month:
                    return -1
                elif self.month>d.month:
                    return 1
                else:
                    if self.day<d.day:
                        return -1
                    elif self.day>d.day:
                        return 1
                    else:
                        return 0


    class MyTime:
        def __init__(self,hour,mimute,second):
            if hour>=0 and hour<24 and mimute>=0 and mimute<60 and second>=0 and
second<60:
                self.hour=hour
                self.minute=mimute
                self.second=second
            else:
                raise Exception("无效时间")

        def show(self,end='\n'):
            print("%02d:%02d:%02d"%(self.hour,self.minute,self.second),end=end)

        def toString(self):
            return "%02d:%02d:%02d"%(self.hour,self.minute,self.second)

        def toSeconds(self):
            return self.hour*3600+self.minute * 60 + self.second
```

```
    @staticmethod
    def fromString(string):
        h,m,s=string.split(":")
        return MyTime(int(h),int(m),int(s))

    def compare(self,t):
        a=self.toSeconds()
        b=t.toSeconds()
        if a>b:
            return 1
        elif a<b:
            return -1
        else:
            return 0

class MyDateTime(MyDate,MyTime):
    def __init__(self,year,month,day,hour,minute,second):
        MyDate.__init__(self,year,month,day)
        MyTime.__init__(self,hour,minute,second)

    def show(self,end='\n'):
        MyDate.show(self,end=' ')
        MyTime.show(self,end=end)

    def toString(self):
        return MyDate.toString(self) + " " + MyTime.toString(self)

    @staticmethod
    def fromString(s):
        a,b=s.split(" ")
        date=MyDate.fromString(a)
        time=MyTime.fromString(b)
        return MyDateTime(date.year,date.month,date.day,
                          time.hour,time.minute,time.second)

    def compare(self,dt):
        if MyDate.compare(self,dt)==0:
            return MyTime.compare(self,dt)
        else:
            return MyDate.compare(self,dt)
try:
    a=MyDateTime(2024,6,30,23,59,50)
    b=MyDateTime.fromString("2024-05-19 10:00:06")
```

```
        a.show()
        b.show()
        sa=a.toString()
        sb=b.toString()
        c=a.compare(b)
        if c==0:
            print(sa,"is equal to ",sb)
        elif c==-1:
            print(sa,"is earlier than ",sb)
        else:
            print(sa,"is later than ",sb)
    except Exception as e:
        print(e)
```

5.5 实践项目 学生信息管理

```
class Student:
    def __init__(self,No,Name,Gender,Age):
        self.No=No
        self.Name=Name
        self.Gender=Gender
        self.Age=Age

    def show(self):
        print("%-16s %-16s %-8s %-4d" %(self.No,self.Name,self.Gender,self.Age))

class StudentList:
    def __init__(self):
        self.students=[]

    def show(self):
        print("%-16s %-16s %-8s %-4s" %("No","Name","Gender","Age"))
        for s in self.students:
            s.show()

    def __insert(self,s):
        i=0
        while (i<len(self.students) and s.No>self.students[i].No):
            i=i+1
        if (i<len(self.students) and s.No==self.students[i].No):
            print(s.No+" 已经存在")
```

```
                return False
        self.students.insert(i,s)
        print("增加成功")
        return True

    def __update(self,s):
        flag=False
        for i in range(len(self.students)):
            if (s.No==self.students[i].No):
                self.students[i].Name=s.Name
                self.students[i].Gender = s.Gender
                self.students[i].Age = s.Age
                print("修改成功")
                flag=True
                break
        if(not flag):
            print("没有这个学生")
        return flag

    def __delete(self,No):
        flag=False
        for i in range(len(self.students)):
            if (self.students[i].No == No):
                del self.students[i]
                print("删除成功")
                flag=True
                break
        if (not flag):
            print("没有这个学生")
        return flag

    def delete(self):
        No=input("No=")
        if(No!=""):
            self.__delete(No)


    def insert(self):
        No=input("No=")
        Name=input("Name=")
        while True:
            Gender=input("Gender=")
            if(Gender=="男" or Gender=="女"):
                break
```

```
                else:
                    print("Gender is not valid")
            Age=input("Age=")
            if(Age==""):
                Age=0
            else:
                Age=int(Age)
            if No!="" and Name!="":
                self.__insert(Student(No,Name,Gender,Age))
            else:
                print("学号、姓名不能为空")

        def update(self):
            No=input("No=")
            Name=input("Name=")
            while True:
                Gender=input("Gender=")
                if(Gender=="男" or Gender=="女"):
                    break
                else:
                    print("Gender is not valid")
            Age=input("Age=")
            if(Age==""):
                Age=0
            else:
                Age=int(Age)
            if No!="" and Name!="":
                self.__update(Student(No,Name,Gender,Age))
            else:
                print("学号、姓名不能为空")

        def process(self):
            while True:
                s = input(">")
                if (s == "show"):
                    self.show()
                elif (s == "insert"):
                    self.insert()
                elif (s == "update"):
                    self.update()
                elif (s == "delete"):
                    self.delete()
                elif (s == "exit"):
                    break
```

```
                else:
                    print("show:      show students")
                    print("insert: insert a new student")
                    print("update: insert a new student")
                    print("delete: delete a student")
                    print("exit:      exit")

st=StudentList()
st.process()
```

第 6 章 Python 文件操作

例 6-1-1：把一个字符串存放在文件中。

```
try:
    fobj=open("c:\\abc.txt","wt")
    fobj.write("abcxyz")
    fobj.close()
except Exception as err:
    print(err)
```

例 6-1-2：打开 abc.txt 文件追加写入另外一个字符串。

```
try:
    fobj=open("c:\\abc.txt","at")
    fobj.write("\nmore")
    fobj.close()
except Exception as err:
    print(err)
```

6.1.5 【案例】学生信息存储到文件

```
def getStudent(i):
    print("输入第", i, "个学生信息")
    try:
        Name = input("姓名:")
        if Name.strip() == "":
            raise Exception("无效的姓名")
        Gender = input("性别:")
        if Gender != "男" and Gender != "女":
            raise Exception("无效的性别")
        Age = input("年龄:")
        Age = float(Age)
        if Age < 18 or Age > 30:
            raise Exception("无效的年龄")
        s={}
        s["Name"]=Name
        s["Gender"]=Gender
        s["Age"]=Age
        return s
    except Exception as err:
        print(err)
        return None

i=1
```

```
try:
    fobj=open("students.txt","wt")
    while True:
        s=getStudent(i)
        if s:
            fobj.write(s["Name"] + "\n" + s["Gender"] + "\n" + str(s["Age"]) + "\n")
            i=i+1
        s=input("继续输入吗(Y/N)")
        if s!="Y" and s!="y":
            break
    fobj.close()
except Exception as err:
    print(err)
```

6.2 读文本文件

例 6-2-1：保存文件 c:\abc.txt，打开文件读取全部内容，把其内容显示在屏幕上。

```
def writeFile():
    fobj=open("c:\\abc.txt","wt")
    fobj.write("abc\nxyz")
    fobj.close()

def readFile():
    fobj=open("c:\\abc.txt","rt")
    s=fobj.read()
    print(s)
    fobj.close()

try:
    writeFile()
    readFile()
except Exception as err:
    print(err)
```

例 6-2-2：保存文件 c:\abc.txt，打开文件读取部分内容，把其内容显示在屏幕上。

```
def writeFile():
    fobj=open("c:\\abc.txt","wt")
    fobj.write("abc\nxyz")
    fobj.close()

def readFile(n):
    fobj=open("c:\\abc.txt","rt")
```

```
        s=fobj.read(n)
        print(s)
        fobj.close()

try:
        writeFile()
        n=3
        print(n)
        readFile(n)
 except Exception as err:
       print(err)
```

例 6-2-3：保存文件 c:\abc.txt，打开文件一次读一个字符，读取全部。

```
def writeFile():
        fobj=open("c:\\abc.txt","wt")
        fobj.write("ab 文 c\n 件 x")
        fobj.close()

def readFile():
        fobj=open("c:\\abc.txt","rt")
        st=""
        while True:
                s=fobj.read(1)
                if s!="":
                        print("read:",ord(s))
                        st=st+s
                else:
                        break
        fobj.close()
        print(st)

try:
        writeFile()
        readFile()
except Exception as err:
        print(err)
```

例 6-2-4：写入 abc 与 xyz 两行，读出显示。

```
def writeFile():
        fobj=open("c:\\abc.txt","wt")
        fobj.write("abc\nxyz")
        fobj.close()

def readFile():
```

```
    fobj=open("c:\\abc.txt","rt")
    s=fobj.readline()
    print(s,"length=",len(s))
    s=fobj.readline()
    print(s,"length=",len(s))
    s=fobj.readline()
    print(s,"length=",len(s))
    fobj.close()

try:
    writeFile()
    readFile()
except Exception as err:
    print(err)
```

例 6-2-5：保存文件 c:\abc.txt，打开文件一次读一行字符，读取全部。

```
def writeFile():
    fobj=open("c:\\abc.txt","wt")
    fobj.write("abc\nxyz")
    fobj.close()

def readFile():
    fobj=open("c:\\abc.txt","rt")
    goon=1
    st=""
    while goon==1:
        s=fobj.readline()
        if s!="":
            st=st+s
        else:
            goon=0
    fobj.close()
    print(st)

try:
    writeFile()
    readFile()
except Exception as err:
    print(err)
```

例 6-2-6： 读取文本文件

```
def writeFile():
    fobj = open("c:\\abc.txt", "wt")
    fobj.write("abc\n 我们\nxyz")
```

```
        fobj.close()

def readFile():
        fobj = open("c:\\abc.txt", "rt")
        for x in fobj.readlines():
                print(x,end='')
        fobj.close()

try:
        writeFile()
        readFile()
except Exception as err:
        print(err)
```

6.2.3 【案例】从文件读出学生信息

1、案例描述

```
class Student:
        def __init__(self,name,gender,age):
                self.name=name
                self.gender=gender
                self.age=age
        def show(self):
                print(self.name,self.gender,self.age)

students=[]
try:
        f=open("student.txt","rt")
        while True:
                name=f.readline().strip("\n")
                if name=="":
                        break
                gender= f.readline().strip("\n")
                age=float(f.readline().strip("\n"))
                students.append(Student(name,gender,age))
        f.close()
        for s in students:
                s.show()
except Exception as err:
        print(err)
```

6.3 二进制文件

例 6-3-1：读写二进制文件

```
def writeFile():
    fobj = open("abc.dat","wb")
    data=b"abc"
    fobj.write(data)
    fobj.close()

def readFile():
    fobj = open("abc.dat","rb")
    data = fobj.read()
    fobj.close()
    print(data)
    print("length of data is:",len(data))

writeFile()
readFile()
```

例 6-3-2：汉字二进制文件存储

```
def writeFile():
    print("Writing to file")
    fobj = open("abc.dat","wb")
    s="二进制"
    data=s.encode("utf-8")
    print(data)
    fobj.write(data)
    fobj.close()

def readFile():
    print("Reading from file")
    fobj = open("abc.dat","rb")
    data = fobj.read()
    print(data)
    s=data.decode("utf-8")
    print(s)
    fobj.close()

writeFile()
readFile()
```

例 6-3-3：一个文件 abc.txt，把它复制到 xyz.txt

```
import os
```

```
def copy(src, dest):
    if os.path.isfile(src):
        print("复制文件:", src, "到", dest)
        fs=open(src, "rb")
        fd=open(dest, "wb")
        while True:
            data=fs.read(1024*1024)
            if not data:
                break
            fd.write(data)
            print("正在复制",len(data),"字节")
        fs.close()
        fd.close()
    else:
        print(src,"文件不存在")

copy("abc.txt", "xyz.txt")
```

例 6-3-4：有 a.txt 与 b.txt 两个文本文件，要求把 b.txt 合并到 a.txt 中。

```
import os

def append(sourceFile,otherFile):
    if os.path.isfile(sourceFile) and os.path.isfile(otherFile):
        fs=open(sourceFile, "ab")
        fo=open(otherFile, "rb")
        while True:
            data=fo.read(1024*1024)
            if not data:
                break
            fs.write(data)
        fs.close()
        fo.close()
        print("文件合并成功")
    else:
        print(sourceFile+"或者"+otherFile+"文件不存在")

append("abc.txt", "xyz.txt")
```

6.3.3 【案例】查看二进制文件数据

```
import os

def peekFile(fileName):
```

```
    if os.path.isfile(fileName):
        fobj= open(fileName, "rb")
        size=0
        while True:
            data=fobj.read(1024*1024)
            if not data:
                break
            size+=len(data)
            for i in range(len(data)):
                b=data[i]
                if b>=32 and b<=126:
                    print(chr(b), end="")
                else:
                    print("." , end="")
                if (i+1)%80==0:
                    print()
        fobj.close()
        print()
        print("文件大小：", size, "字节")
    else:
        print(fileName+"文件不存在")

fileName=input("请输入要查看的文件名：")
peekFile(fileName)
```

6.4 文件编码

例 6-4-1：字符串不同编码转的二进制数据

```
s="二进制"
utf8=s.encode("utf-8")
gbk=s.encode("gbk")
print(utf8)
print(gbk)
print(utf8.decode("utf-8"))
print(gbk.decode("gbk"))
```

例 6-4-2：文本文件不同编码

```
def writeFile():
    fobj= open("abc.txt", "wt",encoding="utf-8")
    fobj.write("你好，世界！\n")
```

```
    fobj.close()

def utf8_readFile():
    fobj= open("abc.txt", "rt",encoding="utf-8")
    data=fobj.read()
    print(data)
    fobj.close()

def gbk_readFile():
    fobj= open("abc.txt", "rt",encoding="gbk")
    data=fobj.read()
    print(data)
    fobj.close()

writeFile()
utf8_readFile()
gbk_readFile()
```

例 6-4-3：文本文件的二进制数据

```
def writeTextFile():
    fobj = open("c:\\abc.txt", "wt",encoding="utf-8")
    fobj.write("abc 我们")
    fobj.close()

def writeBiaryFile():
    fobj = open("c:\\abc.bin", "wb")
    fobj.write("abc 我们".encode("utf-8"))

def seekFile(fileName):
    print(fileName, " contents:")
    fobj = open(fileName, "rb")
    data=fobj.read()
    for i in range(len(data)):
        print(hex(data[i]),end=" ")
    print()
    fobj.close()

def readTextFile():
    fobj = open("c:\\abc.txt", "rt",encoding="utf-8")
    data=fobj.read()
    print(data)
    fobj.close()
```

```
def readBiaryFile():
    fobj = open("c:\\abc.bin", "rb")
    data=fobj.read()
    print(data.decode("utf-8"))
    fobj.close()
try:
    writeTextFile()
    writeBiaryFile()
    seekFile("c:\\abc.txt")
    seekFile("c:\\abc.bin")
    readTextFile()
    readBiaryFile()
except Exception as err:
    print(err)
```

例 6-4-4: Windows 系统文本文件读写回车"\r"字符与换行"\n" 字符

```
def writeTextFile():
    fobj=open("c:\\abc.txt","wt",encoding="utf-8")
    s="a\rb\nc\r\nd\n\re"
    print("writing length:",len(s))
    fobj.write(s)
    fobj.close()
    print("Binary data length:",len(s.encode("utf-8")))

def appendTextFile():
    fobj=open("c:\\abc.txt","ab")
    fobj.write(b"\nf")
    fobj.close()

def seekFile():
    print("contents:")
    fobj = open("c:\\abc.txt", "rb")
    data=fobj.read()
    for i in range(len(data)):
        print(hex(data[i]),end=" ")
    print()
    print("Contents length:",len(data))
    fobj.close()

def show(who,s):
    print(who,"length:",len(s))
    for i in range(len(s)):
        print(hex(ord(s[i])),end=" ")
    print()
```

```
def readTextFile():
    fobj=open("c:\\abc.txt","rt",encoding="utf-8")
    s=fobj.read()
    fobj.close()
    show("readTextFile",s)

def readBinaryFile():
    fobj=open("c:\\abc.txt","rb")
    data=fobj.read()
    fobj.close()
    s=data.decode("utf-8")
    show("readBinaryFile",s)

writeTextFile()
appendTextFile()
seekFile()
readTextFile()
readBinaryFile()
```

6.4.3 【案例】二进制模式读写文本文件

```
def writeFileA():
    fobj = open("c:\\abc.txt", "wb")
    fobj.write("abc 我们".encode("gbk"))
    fobj.close()

def writeFileB():
    fobj = open("c:\\xyz.txt", "wt")
    fobj.write("abc 我们")
    fobj.close()

def readFile(fileName):
    fobj = open(fileName, "rb")
    data=fobj.read()
    for i in range(len(data)):
        print(hex(data[i]),end=" ")
    print()
    fobj.close()

try:
    writeFileA()
    writeFileB()
    readFile("c:\\abc.txt")
```

```
        readFile("c:\\xyz.txt")
except Exception as err:
        print(err)
```

6.5 文件指针

例 6-5-1：文件指针

```
        fobj = open("c:\\abc.txt", "wt")
        print(fobj.tell())
        fobj.write("abc")
        print(fobj.tell())
        fobj.write("我们")
        print(fobj.tell())
        fobj.close()
```

例 6-5-2：读写文件

```
def writeFile():
        fobj = open("c:\\abc.txt", "wt+")
        print(fobj.tell())
        fobj.write("123")
        print(fobj.tell())
        fobj.seek(2,0)
        print(fobj.tell())
        fobj.write("abc")
        print(fobj.tell())
        fobj.close()

def readFile():
        fobj = open("c:\\abc.txt", "rt+")
        rows=fobj.read()
        print(rows)
        fobj.close()

try:
        writeFile()
        readFile()
except Exception as err:
        print(err)
```

例 6-5-3：调整文件指针读写文件

```
def writeFile():
    fobj = open("c:\\abc.txt", "wt+")
    print(fobj.tell())
    fobj.write("123")
    print(fobj.tell())
    fobj.seek(2,0)
    print(fobj.tell())
    fobj.write("abc")
    print(fobj.tell())
    fobj.close()

def readFile():
    fobj = open("c:\\abc.txt", "rt+")
    fobj.write("我们")
    fobj.seek(0,0)
    rows=fobj.read()
    print(rows)
    fobj.close()

try:
    writeFile()
    readFile()
except Exception as err:
    print(err)
```

6.5.4 【案例】文件数据反转与加密

```
import os

def reverseData(data):
    baData=bytearray(data)
    for i in range(len(baData)):
        baData[i]=(255-baData[i])
    data=bytes(baData)
    return data

def reverseFile(fileName):
    fobj=open(fileName,"rb+")
    #文件当前指针位置
    offset=0
    while True:
        data=fobj.read(1024*1024)
        if not data:
            break
        data=reverseData(data)
```

```
            #将文件指针位置移到 offset 处，写入 data 数据
            fobj.seek(offset)
            fobj.write(data)
            #文件指针位置偏移 len(data)
            offset+=len(data)
        fobj.close()

fileName=input("输入文件名: ")
if  os.path.isfile(fileName):
    reverseFile(fileName)
    print("文件反转成功")
else:
    print("文件不存在")
```

6.6 实践项目 教材记录管理

```
class Book:
    def __init__(self, ISBN, Title, Author, Publisher):
        self.ISBN = ISBN
        self.Title = Title
        self.Author = Author
        self.Publisher = Publisher

    def show(self):
        print("%-16s %-16s %-16s %-16s" % (self.ISBN, self.Title, self.Author,
self.Publisher))


class BookList:
    def __init__(self):
        self.books = []

    def show(self):
        print("%-16s %-16s %-16s %-16s" % ("ISBN", "Title", "Author", "Publisher"))
        for s in self.books:
            s.show()

    def __insert(self, s):
        i = 0
        while (i < len(self.books) and s.ISBN > self.books[i].ISBN):
            i = i + 1
        if (i < len(self.books) and s.ISBN == self.books[i].ISBN):
```

```
                print(s.ISBN + "  已经存在")
                return False
        self.books.insert(i, s)
        print("增加成功")
        return True

    def __update(self, s):
        flag = False
        for i in range(len(self.books)):
            if (s.ISBN == self.books[i].ISBN):
                self.books[i].Title = s.Title
                self.books[i].Author = s.Author
                self.books[i].Publisher = s.Publisher
                print("修改成功")
                flag = True
                break
        if (not flag):
            print("没有这个教材")
        return flag

    def __delete(self, ISBN):
        flag = False
        for i in range(len(self.books)):
            if (self.books[i].ISBN == ISBN):
                del self.books[i]
                print("删除成功")
                flag = True
                break
        if (not flag):
            print("没有这个教材")
        return flag

    def delete(self):
        ISBN = input("ISBN=")
        if (ISBN != ""):
            self.__delete(ISBN)

    def insert(self):
        ISBN = input("ISBN=")
        Title = input("Title=")
        Author = input("Author=")
        Publisher = input("Publisher=")
        if ISBN != "" and Title != "":
            self.__insert(Book(ISBN, Title, Author, Publisher))
```

```
        else:
            print("ISBN、教材名称不能空")

    def update(self):
        ISBN = input("ISBN=")
        Title = input("Title=")
        Author=input("Author=")
        Publisher = input("Publisher=")
        if ISBN != "" and Title != "":
            self.__update(Book(ISBN, Title, Author, Publisher))
        else:
            print("ISBN、教材名称不能空")

    def save(self):
        try:
            f = open("books.txt", "wt")
            for b in self.books:
                f.write(b.ISBN + "\n")
                f.write(b.Title + "\n")
                f.write(b.Author+ "\n")
                f.write(b.Publisher+"\n")
            f.close()
        except Exception as err:
            print(err)

    def read(self):
        self.books=[]
        try:
            f = open("books.txt", "rt")
            while True:
                ISBN = f.readline().strip("\n")
                Title = f.readline().strip("\n")
                Author = f.readline().strip("\n")
                Publisher = f.readline().strip("\n")
                if ISBN != "" and Title!="" and Author!="" and Publisher!="":
                    b = Book(ISBN,Title,Author,Publisher)
                    self.books.append(b)
                else:
                    break
            f.close()
        except:
            pass

    def process(self):
```

```
        self.read()
        while True:
            s = input(">")
            if (s == "show"):
                self.show()
            elif (s == "insert"):
                self.insert()
            elif (s == "update"):
                self.update()
            elif (s == "delete"):
                self.delete()
            elif (s == "exit"):
                break
            else:
                print("show:      show Books")
                print("insert: insert a new Book")
                print("update: insert a new Book")
                print("delete: delete a Book")
                print("exit:      exit")
        self.save()


books = BookList()
books.process()
```

第 7 章 Python 数据库操作

7.1 连接 MySQL 数据库

例 7-1-1：Python 连接 MySQL 的 MyDB 数据库

```
import pymysql
try:
    con = pymysql.connect(host="127.0.0.1", port=3306, user="root", passwd="123456", db="mydb", charset="utf8")
    print("连接成功")
    con.close()
except Exception as err:
    print(err)
```

例 7-1-2：在 MySQL 的 MyDB 数据库中创建一张学生记录表 students，包含学号 pNo, 姓名 pName，性别 pGender，年龄 pAge 字段。

```
import pymysql
sql="""
create table students
 (
    pNo varchar(16) primary key,
    pName varchar(16),
    pGender varchar(8),
    pAge int
 )
"""
con = pymysql.connect(host="127.0.0.1", port=3306, user="root", passwd="123456", db="mydb", charset="utf8")
cursor = con.cursor(pymysql.cursors.DictCursor);
try:
    cursor.execute(sql)
    print("done")
except Exception as err:
    print(err)
cursor.close()
```

7.2 读写 MySQL 数据库

例 7-2-1：读取 students 表的记录

```
import pymysql
try:
    con = pymysql.connect(host="127.0.0.1", port=3306, user="root", passwd="123456", db="mydb", charset="utf8")
    cursor = con.cursor(pymysql.cursors.DictCursor)
    cursor.execute("select * from students")
    while True:
        row=cursor.fetchone()
        print(row)
        if not row:
            break
    con.close()
except Exception as err:
    print(err)
结果：
{'pNo': '1', 'pName': 'A', 'pGender': '男', 'pAge': 20}
{'pNo': '2', 'pName': 'B', 'pGender': '女', 'pAge': 21}
None
```

例 7-2-2：读取 students 表的记录各个字段值

```
import pymysql
try:
    con = pymysql.connect(host="127.0.0.1", port=3306, user="root", passwd="123456", db="mydb", charset="utf8")
    cursor = con.cursor(pymysql.cursors.DictCursor)
    cursor.execute("select * from students")
    while True:
        row=cursor.fetchone()
        if not row:
            break
        print(row["pName"],row["pGender"],row["pAge"])
    con.close()
except Exception as err:
    print(err)
```

例 7-2-3：fetchall()读取 students 表的全部记录

```
import pymysql
try:
    con = pymysql.connect(host="127.0.0.1", port=3306, user="root", passwd="123456", db="mydb", charset="utf8")
```

```
        cursor = con.cursor(pymysql.cursors.DictCursor)
        cursor.execute("select * from students")
        rows=cursor.fetchall()
        for row in rows:
            print(row["pName"],row["pGender"],row["pAge"])
        con.close()
    except Exception as err:
        print(err)
```

7.2.3 【案例】学生数据表的管理

```
    import pymysql
    try:
        con = pymysql.connect(host="127.0.0.1", port=3306, user="root", passwd="123456",
db="mydb", charset="utf8")
        cursor = con.cursor(pymysql.cursors.DictCursor)
        cursor.execute("delete from students")
        print(cursor.rowcount)
        cursor.execute("insert into students (pNo,pName,pGender,pAge) values ('1','A','男',20)")
        print(cursor.rowcount)
        cursor.execute("insert into students (pNo,pName,pGender,pAge) values ('2','B','女',21)")
        print(cursor.rowcount)
        cursor.execute("update students set pName='X' where pNo='1'")
        print(cursor.rowcount)
        cursor.execute("update students set pName='X' where pNo='3'")
        print(cursor.rowcount)
        cursor.execute("delete from students where pNo='1'")
        print(cursor.rowcount)
        cursor.execute("delete from students where pNo='3'")
        print(cursor.rowcount)
        con.commit()
        con.close()
    except Exception as err:
        print(err)
```

7.3 MySQL 命令参数

7.3.3 【案例】学生数据表的管理

```
    import pymysql

    class StudentDB:
        def open(self):
```

```
            self.con = pymysql.connect(host="127.0.0.1", port=3306, user="root", passwd="123456", db="mydb", charset="utf8")
            self.cursor = self.con.cursor(pymysql.cursors.DictCursor)
            try:
                sql="create table students (pNo varchar(16) primary key,pName varchar(16),pGender varchar(8) ,pAge int)"
                self.cursor.execute(sql)
            except:
                pass

        def close(self):
            self.con.commit()
            self.con.close()

        def clear(self):
            try:
                self.cursor.execute("delete from students")
            except Exception as err:
                print(err)

        def show(self):
            self.cursor.execute("select pNo,pName,pGender,pAge from students")
            print("%-16s%-16s%-8s%-4s" % ("No","Name","Gender","Age"))
            rows=self.cursor.fetchall()
            for row in rows:
                print("%-16s%-16s%-8s%-4d" % (row["pNo"],row["pName"],row["pGender"],row["pAge"]))

        def insert(self,pNo,pName,pGender,pAge):
            try:
                sql="insert into students (pNo,pName,pGender,pAge) values (%s,%s,%s,%s)"
                self.cursor.execute(sql,(pNo,pName,pGender,pAge))
                print(self.cursor.rowcount," row inserted")
            except Exception as err:
                print(err)

        def update(self,pNo,pName,pGender,pAge):
            try:
                sql="update students set pName=%s,pGender=%s,pAge=%s where pNo=%s"
                self.cursor.execute(sql,(pName,pGender,pAge,pNo))
                print(self.cursor.rowcount," row updated")
            except Exception as err:
                print(err)
```

```
        def delete(self, pNo):
            try:
                sql = "delete from students where pNo=%s"
                self.cursor.execute(sql, (pNo,))
                print(self.cursor.rowcount, " row deleted")
            except Exception as err:
                print(err)

db=StudentDB()
db.open()
db.clear()
db.insert("5","E","女",32)
db.show()
db.update("5","X","女",30)
db.show()
db.insert("2","B","女",20)
db.show()
db.delete("2")
db.show()
db.close()
```

7.4 Sqlite 数据库操作

例 7-4-1：Python 连接 Sqlie 数据库

```
import sqlite3
try:
    con=sqlite3.connect("students.db")
    print("连接成功")
    con.close()
except Exception as err:
    print(err)
```

例 7-4-2：建立一个学生数据库 students.db，并创建一张学生记录表 students，包含学号 pNo,姓名 pName，性别 pGender，年龄 pAge 字段。

```
import sqlite3
sql="""
create table students
 (
    pNo varchar(16) primary key,
    pName varchar(16),
    pGender varchar(8),
```

```
        pAge int
    )
    """
    try:
        con=sqlite3.connect("students.db")
        cursor=con.cursor()
        try:
            cursor.execute(sql)
        except:
            cursor.execute("delete from students")
        cursor.execute("insert into students (pNo,pName,pGender,pAge) values ('1','A','男',20)")
        cursor.execute("insert into students (pNo,pName,pGender,pAge) values ('2','B','女',21)")
        cursor.execute("select * from students")
        rows=cursor.fetchall()
        for row in rows:
            print(row[0],row[1],row[2])
        con.commit()
        con.close()
    except Exception as err:
        print(err)
```

7.4.5 【案例】学生数据表的管理

```
    import sqlite3
    try:
        con=sqlite3.connect("students.db")
        cursor = con.cursor();
        cursor.execute("delete from students")
        print(cursor.rowcount)
        cursor.execute("insert into students (pNo,pName,pGender,pAge) values (?,?,?,?)",('1','A','
男',20))
        print(cursor.rowcount)
        cursor.execute("insert into students (pNo,pName,pGender,pAge) values (?,?,?,?)",('2','B','
女',21))
        print(cursor.rowcount)
        cursor.execute("update students set pName=? where pNo=?",('X','1'))
        print(cursor.rowcount)
        cursor.execute("update students set pName=? where pNo=?",('X','3'))
        print(cursor.rowcount)
        cursor.execute("delete from students where pNo=?",('1',))
        print(cursor.rowcount)
        cursor.execute("delete from students where pNo=?",('3',))
        print(cursor.rowcount)
        con.commit()
```

```
            con.close()
        except Exception as err:
            print(err)
```

7.5 实践项目 学生成绩管理

```
    import pymysql
    import os

    class MarkDB:
        def open(self):
            self.con = pymysql.connect(host="127.0.0.1", port=3306, user="root", passwd="123456", db="mydb", charset="utf8")
            self.cursor = self.con.cursor(pymysql.cursors.DictCursor)
            try:
                sql = "create table marks (pNo varchar(16) primary key,pName varchar(16) ,pChinese float,pMath float,pEnglish float)"
                self.cursor.execute(sql)
            except:
                pass

        def close(self):
            self.con.commit()
            self.con.close()

        def clear(self):
            try:
                self.cursor.execute("delete from marks")
            except Exception as err:
                print(err)

        def show(self):
            try:
                self.cursor.execute("select * from marks")
                print("%-16s%-16s%-8s%-8s%-8s" % ("学号", "姓名", "语文", "数学","英语"))
                rows = self.cursor.fetchall()
                for row in rows:
                    print("%-16s%-16s%-8.0f%-8.0f%-8.0f" % (row["pNo"], row["pName"], row["pChinese"], row["pMath"],row["pEnglish"]))
            except Exception as err:
                print(err)
```

```
    def __insert(self, pNo, pName, pChinese, pMath,pEnglish):
        try:
            sql = "insert into marks (pNo,pName,pChinese,pMath,pEnglish) values
(%s,%s,%s,%s,%s)"
            self.cursor.execute(sql, (pNo, pName, pChinese, pMath,pEnglish))
            print(self.cursor.rowcount, " row inserted")
        except Exception as err:
            print(err)

    def __update(self, pNo, pChinese, pMath,pEnglish):
        try:
            sql = "update marks set pChinese=%s,pMath=%s,pEnglish=%s where pNo=%s"
            self.cursor.execute(sql, ( pChinese, pMath, pEnglish, pNo))
            print(self.cursor.rowcount, " row updated")
        except Exception as err:
            print(err)

    def __delete(self, pNo):
        try:
            sql = "delete from marks where pNo=%s"
            self.cursor.execute(sql, (pNo,))
            print(self.cursor.rowcount, " row deleted")
        except Exception as err:
            print(err)

    def enterMark(self,s):
        while True:
            m=input(s)
            try:
                m=float(m)
                if m>=0 and m<=100:
                    break
            except Exception as err:
                print(err)
        return m

    def insert(self):
        pNo=input("学号:").strip()
        pName=input("姓名:").strip()
        if pNo!="" and pName!="":
            pChinese=self.enterMark("语文:")
            pMath = self.enterMark("数学:")
            pEnglish = self.enterMark("英语:")
```

```
                self.__insert(pNo,pName,pChinese,pMath,pEnglish)
            else:
                print("学号、姓名不能空")

        def update(self):
            pNo = input("学号:").strip()
            if pNo != "":
                pChinese = self.enterMark("语文:")
                pMath = self.enterMark("数学:")
                pEnglish = self.enterMark("英语:")
                self.__update(pNo, pChinese, pMath, pEnglish)
            else:
                print("学号不能空")

        def delete(self):
            pNo = input("学号:").strip()
            if pNo != "":
                self.__delete(pNo)
            else:
                print("学号不能空")

        def export(self):
            try:
                f=open("marks.txt","wt",encoding="utf-8")
                self.cursor.execute("select * from marks")
                f.write("学号,姓名,语文,数学,英语\n")
                rows = self.cursor.fetchall()
                for row in rows:

f.write(row["pNo"]+","+row["pName"]+","+str(row["pChinese"])+","+str(row["pMath"])+","+str(r
ow["pEnglish"])+"\n")
                f.close()
                print("成绩导出完毕")
            except Exception as err:
                print(err)

        def __parseMark(self,s):
            m=0
            try:
                m=float(s)
                if m<0 or m>100:
                    m=0
            except:
                pass
```

```
            return m

        def load(self):
            try:
                s=input("输入导入文件路径与名称:")
                if os.path.exists(s):
                    f=open(s,"rt",encoding="utf-8")
                    s=f.readline().strip("\n")
                    st=s.split(",")
                    if len(st)==5 and st[0]=="学号" and st[1]=="姓名" and st[2]=="语文" and
st[3]=="数学" and st[4]=="英语":
                        s="going"
                        while s!="":
                            s=f.readline().strip("\n")
                            if s!="":
                                st=s.split(",")
                                if len(st)==5:
                                    pNo=st[0].strip()
                                    pName=st[1].strip()
                                    pChinese=self.__parseMark(st[2])
                                    pMath = self.__parseMark(st[3])
                                    pEnglish = self.__parseMark(st[3])
                                    if pNo!="" and pName!="":

self.__insert(pNo,pName,pChinese,pMath,pEnglish)
                        print("成绩转载完毕")
                    else:
                        print("文件格式不正确")
                    f.close()
                else:
                    print(s+"文件不存在")
            except Exception as err:
                print(err)

        def process(self):
            self.open()
            while True:
                s=input(">")
                if s=="show":
                    self.show()
                elif s=="insert":
                    self.insert()
                elif s=="update":
                    self.update()
```

```
                elif s=="delete":
                        self.delete()
                elif s=="export":
                        self.export()
                elif s=="load":
                        self.load()
                elif s=="exit":
                        break
                else:
                        print("Accept commands: show/insert/update/delete/export/load/exit")
                        print("show      ---  显示所有记录")
                        print("insert ---    增加一条记录")
                        print("update ---  修改一条记录")
                        print("delete ---    删除一条记录")
                        print("export ---    导出到文件")
                        print("load    ---    从文件导入")
                        print("exit      ---    退出系统")
        self.close()

db=MarkDB()
db.process()
```

第 8 章 Python 网络编程

8.1 网络通信程序

8.1.4 【案例】网络通信程序

(1) 服务器程序

```
import socket
import datetime
try:
    server = socket.socket()
    host = socket.gethostname()
    port = 2345
    server.bind((host, port))
    server.listen()
    print(host,"在监听...")
    client, address = server.accept()
    print(address,'客户端连接')
    data='欢迎,我是服务器'.encode()
    client.send(data)
    while True:
        data=client.recv(1024)
        data=data.decode()
        dt=datetime.datetime.now().strftime('%H:%M:%S')
        print(dt+"  客户端说:",data)
        if data=='bye':
            break
    client.send('服务器说:再见'.encode())
    client.close()
    server.close()
except Exception as e:
    print(e)
```

(2) 客户端程序

```
import socket
input("按任意键开始连接服务器...")
try:
    client = socket.socket()
```

```
        host = socket.gethostname()
        port = 2345
        client.connect((host, port))
        print("连接服务器成功")
        data=client.recv(1024)
        data=data.decode()
        print("服务器说:",data)
        while True:
            data=input("输入要发送的消息:")
            client.send(data.encode())
            if data=="bye":
                break
        data=client.recv(1024)
        data=data.decode()
        print(data)
        client.close()
    except Exception as e:
        print(e)
```

8.2 整数网络传输

8.2.3【案例】整数网络传输

(1) 服务器程序

```
import socket
import struct

def readInt(socket):
    size=struct.calcsize("@i")
    data=socket.recv(size)
    n=struct.unpack("@i",data)[0]
    return n

def writeInt(socket,n):
    data=struct.pack("@i",n)
    socket.send(data)

try:
    server = socket.socket()
    host = socket.gethostname()
    port = 2345
    server.bind((host, port))
```

```
        server.listen()
        print(host,"在监听...")
        client= server.accept()[0]
        print('客户端连接')
        n=readInt(client)
        print(n)
        writeInt(client,n)
        client.close()
        server.close()
except Exception as e:
        print(e)
```

(2) 客户端程序

```
    import socket
    import struct

    def readInt(socket):
        size=struct.calcsize("@i")
        data=socket.recv(size)
        n=struct.unpack("@i",data)[0]
        return n

    def writeInt(socket,n):
        data=struct.pack("@i",n)
        socket.send(data)

    try:
        a=input("输入整数")
        a=int(a)
        client = socket.socket()
        host = socket.gethostname()
        port = 2345
        client.connect((host, port))
        print("连接服务器成功")
        writeInt(client,a)
        b=readInt(client)
        print(b)
        client.close()
    except Exception as e:
        print(e)
```

8.3 字符串网络传输

8.3.3 【案例】字符串网络传输

(1) 服务器程序

```
import socket
import struct

def readString(socket):
    size=struct.calcsize("@i")
    d=socket.recv(size)
    n=struct.unpack("@i",d)[0]
    data=socket.recv(n)
    return data.decode("utf-8")

def writeString(socket,s):
    data=s.encode("utf-8")
    size=len(data)
    d=struct.pack("@i",size)
    socket.send(d)
    socket.send(data)

try:
    server = socket.socket()
    host = socket.gethostname()
    port = 2345
    server.bind((host, port))
    server.listen()
    print(host,"在监听...")
    client= server.accept()[0]
    print('客户端连接')
    s=readString(client)
    print(s)
    writeString(client,s)
    client.close()
    server.close()
except Exception as e:
    print(e)
```

(2) 客户端程序

```
import socket
import struct
```

```
def readString(socket):
        size=struct.calcsize("@i")
        d=socket.recv(size)
        n=struct.unpack("@i",d)[0]
        data=socket.recv(n)
        return data.decode("utf-8")

def writeString(socket,s):
        data=s.encode("utf-8")
        size=len(data)
        d=struct.pack("@i",size)
        socket.send(d)
        socket.send(data)

try:
        s=input("输入字符串")
        client = socket.socket()
        host = socket.gethostname()
        port = 2345
        client.connect((host, port))
        print("连接服务器成功")
        writeString(client,s)
        t=readString(client)
        print(t)
        client.close()
except Exception as e:
        print(e)
```

8.4 实践项目 文件网络传输

1、服务器程序 server.py

```
import socket
import struct
import os

def readInt(socket):
        size=struct.calcsize("@q")
        data=socket.recv(size)
        n=struct.unpack("@q",data)[0]
        return n

def writeInt(socket,n):
```

```
    data=struct.pack("@q",n)
    socket.send(data)

def readString(socket):
    size=struct.calcsize("@i")
    d=socket.recv(size)
    n=struct.unpack("@i",d)[0]
    data=socket.recv(n)
    return data.decode("utf-8")

def writeString(socket,s):
    data=s.encode("utf-8")
    size=len(data)
    d=struct.pack("@i",size)
    socket.send(d)
    socket.send(data)

try:
    if not os.path.exists("uploadFiles"):
        os.mkdir("uploadFiles")
    server = socket.socket()
    host = socket.gethostname()
    port = 2345
    server.bind((host, port))
    server.listen()
    print(host,"在监听...")
    client= server.accept()[0]
    fileName = readString(client)
    print("文件名称:" + fileName)
    fileLength = readInt(client)
    print("文件尺寸:", fileLength)
    fobj=open("uploadFiles/" + fileName, "wb")
    size=1024*1024*2
    recvLength = 0
    while recvLength < fileLength:
        if fileLength-recvLength < size:
            size = fileLength-recvLength
        data = client.recv(size)
        fobj.write(data)
        recvLength += len(data)
        print("已接收:", recvLength, "/", fileLength)
    fobj.close()
    print("上传成功!")
    writeString(client, "OK")
```

```
        client.close()
        server.close()
except Exception as e:
        print(e)
```

2、客户端程序 client.py

```
import socket
import struct
import os
from tkinter import filedialog
import tkinter as tk

def readInt(socket):
        size=struct.calcsize("@q")
        data=socket.recv(size)
        n=struct.unpack("@q",data)[0]
        return n

def writeInt(socket,n):
        data=struct.pack("@q",n)
        socket.send(data)

def readString(socket):
        size=struct.calcsize("@i")
        d=socket.recv(size)
        n=struct.unpack("@i",d)[0]
        data=socket.recv(n)
        return data.decode("utf-8")

def writeString(socket,s):
        data=s.encode("utf-8")
        size=len(data)
        d=struct.pack("@i",size)
        socket.send(d)
        socket.send(data)

try:
        root=tk.Tk()
        root.withdraw()
        fileName = filedialog.askopenfilename(title="选择文件", filetypes=[("所有文件", "*")])
        if os.path.exists(fileName):
                fobj = open(fileName, "rb+")
                fobj.seek(0, 2)
```

```
            fileLength=fobj.tell()
            fobj.seek(0, 0)
            size=1014*1024
            postLengh=0
            p = fileName.rfind("/")
            fileName = fileName[p + 1:]
            print("文件名称:", fileName)
            print("文件尺寸:", fileLength)
            client = socket.socket()
            host = socket.gethostname()
            port = 2345
            client.connect((host, port))
            writeString(client, fileName)
            writeInt(client, fileLength)
            while postLengh < fileLength:
                data = fobj.read(size)
                if data:
                    client.send(data)
                    postLengh += len(data)
                    print("已上传:", postLengh,"/",fileLength)
            fobj.close()
            resp = readString(client)
            if resp == "OK":
                print("上传成功!")
            else:
                print("上传失败!")
            client.close()
        else:
            print(fileName+"不存在!")
        root.destroy()
except Exception as e:
        print(e)
```

第 9 章　Python 综合项目

9.1 所得税计算器

```
m=input("全年应纳税额:")
try:
    m=float(m)
    tax=0
    if m<=36000:
        tax=m*0.03
    elif m<=144000:
        tax=m*0.1-2520
    elif m<=300000:
        tax=m*0.2-16920
    elif m<=420000:
        tax=m*0.25-31920
    elif m<=660000:
        tax=m*0.3-52920
    elif m<=960000:
        tax=m*0.35-85920
    else:
        tax=m*0.45-181920
    print("应缴所得税是:%.2f"%tax)
except Exception as e:
    print("输入错误:",e)
```

9.2 神奇数值变换

1、验证三位数

```
def toList(n):
    d=[0,0,0]
    i=2
    while i>=0:
        d[i]=n%10
        n=n//10
```

```
            i=i-1
        return d

    def process():
        for a in range(0,10):
            for b in range(0,10):
                for c in range(0,10):
                    if not(a==b and a==c):
                        n=a*100+b*10+c
                        print("verify ",n)
                        while n!=495:
                            d=toList(n)
                            d.sort()
                            min=d[0]*100+d[1]*10+d[2]
                            max= d[2] * 100 + d[1] * 10 + d[0]
                            n=max-min
                            print("%03d"%max,"-","%03d"%min,"=","%03d"%n)

    process()
```

2、验证四位数

```
    def toList(n):
        d=[0,0,0,0]
        i=3
        while i>=0:
            d[i]=n%10
            n=n//10
            i=i-1
        return d

    def enter():
        while True:
            try:
                n=input("输入一个数字不完全相同的四位数:")
                n=int(n)
                if n>=1000 and n<=9999:
                    d=toList(n)
                    if d[0]==d[1] and d[0]==d[2] and d[0]==d[3]:
                        print("四位数字不能完全相同")
                    else:
                        break
            except Exception as e:
                print(e)
        return n
```

```
def process():
    n=enter()
    print(n)
    while n!=6174:
        d=toList(n)
        d.sort()
        min=d[0]*1000+d[1]*100+d[2]*10+d[3]
        max= d[3] * 1000 + d[2] * 100 + d[1] * 10 + d[0]
        n=max-min
        print("%04d"%max,"-","%04d"%min,"=","%04d"%n)

process()
```

3、验证五位数

```
#验证程序
def toList(n):
    d=[0,0,0,0,0]
    i=4
    while i>=0:
        d[i]=n%10
        n=n//10
        i=i-1
    return d

def process():
    for a in range(0,10):
        for b in range(0,10):
            for c in range(0,10):
                for d in range(0,10):
                    for e in range(0,10):
                        if not(a==b and a==c and a==d and a==e):
                            n=a*10000+b*1000+c*100+d*10+e
                            print("verify ",n)
                            while n!=71973 and n!=75933 and n!=59994:
                                x=toList(n)
                                x.sort()
                                min=x[0]*10000+x[1]*1000+x[2]*100+x[3]*10+x[4]
                                max=
x[4]*10000+x[3]*1000+x[2]*100+x[1]*10+x[0]
                                n=max-min
                                print("%05d"%max,"-
","%05d"%min,"=","%05d"%n)
```

```
process()
```

9.3 乒乓球比赛名次

```
for a in range(1,6):
    for b in range(1,6):
        for c in range(1,6):
            for d in range(1,6):
                for e in range(1,6):
                    if a!=b and a!=c and a!=d and a!=e and b!=c and b!=d and b!=e and c!=d and c!=e and d!=e and e!=2 and e!=3:
                        if a in [1,2]:
                            ca=1 if e==1 else 0
                        else:
                            ca =1 if e!=1 else 0
                        if b in [1,2]:
                            cb=1 if b==2 else 0
                        else:
                            cb =1 if b!=2 else 0
                        if c in [1,2]:
                            cc=1 if a==5 else 0
                        else:
                            cc =1 if a!=5 else 0
                        if d in [1,2]:
                            cd=1 if c!=1 else 0
                        else:
                            cd =1 if c==1 else 0
                        if e in [1,2]:
                            ce=1 if d==1 else 0
                        else:
                            ce =1 if d!=1 else 0
                        if ca +cb+cc+cd+ce==5:
                            print("a --",a)
                            print("b --",b)
                            print("c --",c)
                            print("d --",d)
                            print("e --",e)
```

9.4 汉字编码表

```
def showAll():
    code=bytearray(b"\x00\x00")
    for i in range(1,95):
        print("---------- ",i,"区 ----------")
        code[0]=i+0xa0
        for j in range(1,95):
            code[1]=j+0xa0
            try:
                c=code.decode("gbk")
            except:
                c=""
            print("%s(%s)  "%(c,code.hex().upper()),end="")
            if j%8==0:
                print()
        print()
        input("Press any key to continue...")

def codeToChar():
    gbk=input("请输入十六进制 GBK 编码：")
    try:
        code=bytearray.fromhex(gbk)
        c=code.decode("gbk")
        print("GBK 编码为：",gbk,"GBK 字符为：",c)
    except:
        print("-----错误")

def charToCode():
    c=input("请输入 GBK 字符：")
    try:
        code=c.encode("gbk").hex().upper()
        print("GBK 字符为：",c,"GBK 编码为：",code)
    except:
        print("-----错误")

while True:
    print()
    print("1.显示所有 GBK 编码")
    print("2.GBK 编码转字符")
    print("3.字符转 GBK 编码")
    print("4.退出程序")
    choice=input("请输入选项：")
```

```
        if choice=="1":
            showAll()
        elif choice=="2":
            codeToChar()
        elif choice=="3":
            charToCode()
        elif choice=="4":
            break
```

9.5 学生成绩分析

```
import math
import random
import matplotlib
import matplotlib.pyplot as plt
import os

def create():
    n=random.randint(50,60)
    fobj=open("marks.csv","wt",encoding="utf-8")
    for i in range(n):
        no="%03d"%i
        name="N"+no
        m=int(random.gauss(75,10))
        chinese=m if m>=0 and m<=100 else 50
        m=int(random.gauss(80,10))
        math=m if m>=0 and m<=100 else 50
        m=int(random.gauss(85,10))
        english=m if m>=0 and m<=100 else 50
        fobj.write("%s,%s,%f,%f,%f\n"%(no,name,chinese,math,english))
    fobj.close()

def load():
    global st
    try:
        fobj=open("marks.csv","rt",encoding="utf-8")
        rows=fobj.readlines()
        for row in rows:
            t=row.strip()
            if t!="":
                t=t.split(",")
                if len(t)==5:
```

```
                    s={}
                    s["no"]=t[0]
                    s["name"]=t[1]
                    s["chinese"]=float(t[2])
                    s["math"]=float(t[3])
                    s["english"]=float(t[4])
                    st.append(s)
            except Exception as err:
                print(err)


    def show(group):
        print("%10s%10s%10s%10s%10s"%("no","name","chinese","math","english"))
        for s in group:

print("%10s%10s%9.1f%9.1f%9.1f"%(s["no"],s["name"],s["chinese"],s["math"],s["english"]))

    def st_sort(course):
        if course in ["chinese","math","english"]:
            return sorted(st,key=lambda s: s[course])

    def st_range(course):
        if course in ["chinese","math","english"]:
            countName=["[0,60)","[60,70)","[70:80)","[80:90)","[90:100)"]
            count=[0 for c in countName]
            for s in st:
                m=s[course]
                if m<60:
                    count[0]+=1
                elif m<70:
                    count[1]+=1
                elif m<80:
                    count[2]+=1
                elif m<90:
                    count[3]+=1
                else:
                    count[4]+=1
            print(cName[course],"分数段人数")
            for i in range(len(countName)):
                print(countName[i],":",count[i])
            plt.subplot(2,1,1)
            plt.pie(count,labels=countName,autopct="%.1f%%")
            plt.title(cName[course]+"分数段百分比例")
            plt.subplot(2,1,2)
```

```
            plt.bar(countName,count)
            plt.title(cName[course]+"分数段人数")
            plt.show()

def maxValue(cm):
    m=cm[0]
    for e in cm:
        if e>m:
            m=e
    return m

def minValue(cm):
    m=cm[0]
    for e in cm:
        if e<m:
            m=e
    return m

def aveValue(cm):
    s=0
    s2=0
    for e in cm:
        s+=e
        s2+=e*e
    s=s/len(cm)
    s2=math.sqrt((s2/len(cm)-s*s))
    return s,s2

def st_stat(course):
    if course in ["chinese","math","english"]:
        cm=[s[course] for s in st]
        max=maxValue(cm)
        min=minValue(cm)
        a,d=aveValue(cm)
        maxs=[]
        mins=[]
        for s in st:
            if s[course]==max:
                maxs.append(s)
            if s[course]==min:
                mins.append(s)
        course=cName[course]
        print(course,"最高分:",max)
        show(maxs)
```

```
            print(course,"最低分:",min)
            show(mins)
            print(course,"平均分:%.2f"%a)
            print(course,"均方差:%.2f"%d)
            print()

matplotlib.rcParams['font.sans-serif']=['SimHei']
matplotlib.rcParams['axes.unicode_minus']=False
cName={"chinese":"语文","math":"数学","english":"英语"}
st=[]
#如果 marks.csv 不存在就生成
if not os.path.exists("marks.csv"):
    create()
load()
show(st)
print("1.语文、2.数学、3.英语，分析哪一科目?")
s=input("输入 1,2,3:")
course=""
if s=="1":
    course="chinese"
elif s=="2":
    course="math"
elif s=="3":
    course="english"
if course!="":
    sortedList=st_sort(course)
    show(sortedList)
    st_stat(course)
    st_range(course)
else:
    print("输入错误")
```

9.6 数字迷宫游戏

```
import random

def seekPath():
    global maze,mark,path,count
    #目前位置(x,y)
    x,y=path[len(path)-1]
```

```
    for dx,dy in [(0,1),(0,-1),(1,0),(-1,0)]:
        #新位置(nx,ny)
        nx=x+dx
        ny=y+dy
        if nx>=0 and    nx<len(maze) and ny>=0 and ny<len(maze[0]):
            if maze[nx][ny]==0 and mark[nx][ny]==0:
                #可以到达(nx,ny)
                if nx==len(maze)-1 and ny==len(maze[0])-1:
                    #到达终点
                    path.append((nx,ny))
                    count+=1
                    print("路径：%2d "%count,path)
                    path.pop()
                else:
                    #继续走到(nx,ny)
                    path.append((nx,ny))
                    mark[nx][ny]=1
                    #递归搜索
                    seekPath()
    #回溯到上一位置
    i,j=path.pop()
    mark[i][j]=0

def show(maze):
    #显示迷宫
    for row in maze:
        for e in row:
            print(e,end=" ")
        print()

def process(demo=False):
    global maze,mark,path,count
    if demo:
        #示例迷宫
        m=3
        n=4
        maze=[[0, 0, 0, 1 ],
        [0, 0, 0, 0],
        [0, 0, 1, 0]
        ]
    else:
        #随机生成迷宫
        m=random.randint(3,6)
        n=random.randint(3,6)
```

```
            maze=[]
            for i in range(m):
                row=[(1 if random.randint(0,3)==0 else 0) for j in range(n)]
                maze.append(row)
            maze[0][0]=0
            maze[m-1][n-1]=0
        print("迷宫:(m=%d,n=%d)"%(m,n))
        show(maze)
        #初始化 mark
        mark=[[0 for j in range(n)] for i in range(m)]
        path=[(0,0)]
        mark[0][0]=1
        count=0
        #开始搜索路径
        seekPath()
        print("路径数量：%d"%count)

process(True)
process()
```

9.7 最短路径问题

```
G=[[0,3,0,2,1],
[3,0,2,0,4],
[0,2,0,5,7],
[2,0,5,0,4],
[1,4,7,4,0] ]

def distance(G,path):
    D=0
    m=len(path)
    for i in range(m-1):
        D+=G[path[i]][path[i+1]]
    return D

def show(kept):
    print("       Distance:%3d"%kept["distance"]," Path: ",end="")
    path=kept["path"]
    m=len(path)
    for i in range(m):
        print(chr(ord("A")+path[i]),end="")
```

```
            if i!=m-1:
                print("-",end="")
        print()

def seek(G,kept,path,j):
    p=path[len(path)-1]
    D=distance(G,path)
    for q in range(len(G)):
        if G[p][q]>0 and q not in path and D+G[p][q]<kept["distance"]:
            path.append(q)
            if q==j:
                if D+G[p][q]<kept["distance"]:
                    kept["distance"]=D+G[p][q]
                    kept["path"]=path.copy()
                path.pop()
            else:
                seek(G,kept,path,j)
    if path:
        path.pop()

n=len(G)
maxD=0
for i in range(n):
    for j in range(n):
        maxD+=G[i][j]
for i in range(n):
    for j in range(i+1,n):
        path=[i]
        kept={"distance":maxD,"path":[]}
        print("Seeking path between",chr(ord("A")+i),"and",chr(ord("A")+j))
        seek(G,kept,path,j)
        show(kept)
```

9.8 背包价值问题

```
def solveKnapsack(TW):
    maxValue=0
    knapsack=[]
    for A in range(2):
        for B in range(2):
            for C in range(2):
```

```
                    for D in range(2):
                        for E in range(2):
                            w=A*ws[0]+B*ws[1]+C*ws[2]+D*ws[3]+E*ws[4]
                            v=A*vs[0]+B*vs[1]+C*vs[2]+D*vs[3]+E*vs[4]
                            if w<=TW and v>maxValue:
                                maxValue=v
                                knapsack=[A,B,C,D,E]
    s=[]
    w=0
    for i in range(len(knapsack)):
        if knapsack[i]==1:
            s.append([chr(ord('A')+i),[ws[i],vs[i]]])
            w+=ws[i]
    print(s)
    print("总重量:",w,"总价值:",maxValue)

ws=[10,6,8,5,12]
vs=[7,9,10,12,6]
W=int(input("输入背包容量: "))
if W<5:
    print("背包容量不足，无解")
else:
    solveKnapsack(W)
```