第3章 面向对象程序设计

主要内容

- 类和对象
- 类的继承
- 多态与重载
- 模块与类

===类和对象===

◆ 定义和使用类

➢ 声明类

【格式】

class 类名：

成员变量

成员函数

➢ 定义类的对象

【格式】

对象名=类名()

➢ 成员变量

公有变量

私有变量 __xxx

```
class person:
    def SayHello(self):
        print("hello class!")
p=person()
p.SayHello()
```

```
class per1:
    str='Myfirst'
    __t='Yes'
    def output(self):
        print(self.str)
        print(self.__t)
def m():
    p=per1()
    print(p.str)
    p.output()
m()
```

===类和对象===

◆定义和使用类

- ➢ 构造函数

  def __init__(self,其他参数):
  语句块

- ➢ 析构函数

  def __del__(self)

```
class MyS:
    def __init__(self):
        self.str='MyString'
        self.__t='Yes1'
    def output(self):
        print(self.__t)
        print(self.str)
```

```
def __del__(self):
    print('bye-bye')
```

```
class MyS:
    def __init__(self,name,pwd):
        self.name=name
        self.__pwd=pwd
    def output(self):
        print(self.name,self.__pwd)
```

◆ 定义和使用类

➢ 静态成员与静态方法

(1) 它们都是属于类的

(2) 静态方法无须传入self参数，无法访问实例变量

(3) 直接通过类名访问

【语法格式】

```
class 类名：
    @staticmethod
    def 静态方法名():
        方法体
```

```
class MyS:
    var1='String 1'
    @staticmethod
    def staticmd():
        print('I am static method!')
```

问题1：何为类的私有变量，它是如何定义的？如何对它进行访问？

问题2：何为类的静态方法，它是如何定义的？如何对它进行访问？

第7章 面向对象程序设计

主要内容

- 类和对象
- 类的继承
- 多态与重载
- 模块与类

===类的继承===

【格式】

class 派生类名(基类名1,基类名2,……):

类体

【例3-1】声明一个公民类，包括身份证号、姓名、年龄，声明学生类、教师类继承于公民类，学生类有学号、班级和成绩，教师类有工号、系别、薪水

```
class C:
    def __init__(self,id,name,age):
        self.id=id
        self.name=name
        self.age=age
    def __del__(self):
        print('bye')
```

===类的继承===

【例3-1】声明一个公民类，声明学生类、教师类

子类信息

```
class S(C):
    def __init__(self,id,name,age,stdno,grade,score):
        super(S,self).__init__(id,name,age)
        self.stdno=stdno
        self.grade=grade
        self.score=score
```

```
class T(C):
    def __init__(self,id,name,age,Thno,dept,sal):
        super(T,self).__init__(id,name,age)
        self.Thno=Thno
        self.dept=dept
        self.sal=sal
```

===类的继承===

【例3-1】声明一个公民类，声明学生类、教师类

➢ 类的使用—属性访问

```
if __name__=='__main__':
    c=C('01','张三疯',65)
    print(c.id,c.name,c.age)
    del c
    s=S('02','张无忌',18,'160400101',1,95)
    print(s.id,s.name,s.age,s.stdno,s.grade,s.score)
    del s
    t=T('01','张cuishan',40,’0101022','computer',6000)
    print(t.id,t.name,t.age,t.Thno,t.dept,t.sal)
```

===类的继承===

➢ 派生类和基类的同名方法

【格式】 class 派生类名(基类名1,基类名2,……):
与父类同名的子类方法(参数列表):
super(子类).基类同名方法
子类其语句

```
class C:
    ……
  def show(self):
     print(self.id,self.name,self.age,end=" ")
```

```
class T(C):
……
   def show(self):
       super(T,self).show()
       print(self.Thno,self.dept,self.sal,end=" ")
```

===类的继承===

- 派生类和基类的同名方法

【例3-1】声明一个公民类，声明学生类、教师类

- 类的使用—方法调用

```
c=C('01','张三疯',65)
c.show()
del c
t=T('02','张无忌',28,'0400101','computer',6000)
t.show()
del t
```

问题：类的继承的意义是什么，它的设计要点是什么？需要注意哪些方面？

第3章 面向对象程序设计

主要内容

- 类和对象
- 类的继承
- 多态与重载
- 模块与类

➢ 抽象类和多态

【定义抽象类格式】

```
from abc import ABCMeta,abstractmethod
class myAbc(object):
        __metaclass__=ABMeta
       @abstractmethod
        def abstractmethod(self):pass
```

【例3-2】定义一个画图类的基本框架，抽象基类包括公共颜色和抽象方法draw，子类线包括起点与终点、实现抽象方法draw,子类圆包括圆心与半径。

===多态与重载===

```
from abc import
ABCMeta,abstractmethod
class S(object):
    __metaclass__=ABCMeta
    def __init__(self):
        self.color='black'
    @abstractmethod
    def draw(self):pass
```

```
class Line(S):
    def __init__(self,x1,y1,x2,y2):
        self.x1=x1
        self.y1=y1
        self.x2=x2
        self.y2=y2
    def draw(self):
        print('Draw line:(%d, %d, %d,
%d)'%(self.x1,self.y1,self.x2,self.y2))
```

```
class C(S):
    def __init__(self,x,y,r):
        self.x=x
        self.y=y
        self.r=r
    def draw(self):
        print('Draw circle:(%d,
%d, %d)'%(self.x,self.y,self.r))
```

```
def f():
    c=C(10,10,5)
    l=Line(5,5,15,15)
    lst=[]
    lst.append(c)
    lst.append(l)
    for k in range(len(lst)):
        lst[k].draw()
```

问题：为什么要有抽像类，它的设计要点是什么？

===多态与重载===

◆ 构造函数的重载

➢ 类成员与类方法

(1) 直接通过类名访问

(2) 构成类的多形态对象

【语法格式】

```
class 类名：
    @classmethod
    def 类方法名([cls[,参数列表]]):
        方法体
```

```
import time
class zDate:
    def __init__(self, year, month, day):
        self.year = year
        self.month = month
        self.day =day
    @classmethod
    def today(cls):
        t = time.localtime()
        return cls(t.tm_year, t.tm_mon,
                   t.tm_mday)
```

类自身

```
>>> a = zDate(2012,7,2)
>>> b = zDate.today()
>>> print(a.year, a.month, a.day)
>>> print(b.year, b.month, b.day)
```

➢ 运算符重载

方法名	运算符和表达式	说明
__add__(self, rhs)	self + rhs	加法
__sub__(self, rhs)	self - rhs	减法
__mul__(self, rhs)	self * rhs	乘法
__truediv__(self, rhs)	self / rhs	除法
__floordiv__(self, rhs)	self // rhs	整除
__mod__(self, rhs)	self % rhs	求余
__pow__(self, rhs)	self ** rhs	求幂运算

===多态与重载===

➢ 运算符重载

【例3-3】定义一个矢量类，包括x和y实例成员变量、构造方法以及两个矢量的加法、减法和乘法的运算符

```
class Vector2:
    def __init__(self,x,y):
        self.x=x
        self.y=y
    def __add__(self,v):
        return self.x+v.x,self.y+v.y
    def __sub__(self,v):
        return self.x-v.x,self.y-v.y
    def __mul__(self,v):
        return self.x*v.x,self.y*v.y
```

```
def f():
    v1=Vector2(1,2)
    v2=Vector2(3,4)
    print(v1+v2)
    print(v2*v1)
f()
```

===多态与重载===

➢ 运算符重载

【例3-4】定义一个复数类，包括实部和虚部实例成员变量、构造方法以及两个复数的加法、乘法的运算符

```
class compl:
    def __init__(self,r,i):
        self.r=r
        self.i=i
```

```
def show(self):
        print(self.r,"+",self.i,'j')
```

```
def __add__(self,c):
        return compl(self.r+c.r,self.i+c.i)
```

```
def __mul__(self,c):
        return compl(self.r*c.r-
                self.i*c.i,self.r*c.i+c.r*self.i)
```

```
def f():
    c1=compl(3,4)
    c2=compl(6,-7)
    c3=c1+c2
    c4=c1*c2
    c4.show()
    (c1+c2).show()
```

第3章 面向对象程序设计

主要内容

- 类和对象
- 类的继承
- 多态与重载
- 模块与类

==模块与类==

◆模块

文件名.py是模块，一个类一个块

(1) 引入类

import 模块名 或 from 模块名 import 类名

如，a.py中义了A类，在类C中引入了A

import a 或 from a import A

(2) 包

一个包含__init__.py的文件夹，称为包

(3)引入包中的模块

import 包名.模块名

模块名.类名

===习题===

(1)设计一个日期类，它包括年、月、日三个实例成员变量，其中年设计为私有的，编写构造方法、年月日的显示方法及修改年值的方法，最后编写主模块定义其对象，赋值为当前日期，对对象的值进行修改并显示对象结果。

(2)编写一个圆类，它包括表示半径的变量、构造方法、修改半径的方法、显示半径的方法以及计算圆面积的方法；然后继承圆类再编写一个圆柱体派生类，它包括表示高度的变量、构造方法、修改半径和高度的方法、显示半径和高度的方法以及计算圆柱体体积的方法。最后编写主模块定义这两个类的对象，并进行适当的赋值，对对象的值进行修改并显示对象的结果

===习题===

(4)一个列表由若干整数构成，编写函数删除其中素数元素。
(5)编写函数，求两个正整数的最小公倍数。
(6)录入二个学生的成绩，把该学生的成绩转换成A：优秀、B：良好、C：合格、D：及格的形式，最后将该学生的成绩打印出来。要求使用assert断言处理分数不合理的情况。
(7)现在许多显示器的屏幕宽度和高度的比例是16:9。讨论显示器的尺寸讲的是对角线长度，单位是英寸。现编写程序，输人显示器的尺寸，单位英寸，计算并输出显示器的宽度和高单位为厘米。1英寸（in）=2.54厘米（cm）

本章小结

- 类和对象
- 继承
- 多态与重载
- 模块与类