# -\*- coding: utf-8 -\*-

class Employee:

def \_\_init\_\_(self,name,age,salary):#初始化

print('Created a class:',self.\_\_class\_\_)#打印类名字

self.name=name

self.age=age

self.salary=salary

def getter\_name(self):

return self.name

def getter\_age(self):

return self.age

def getIncome(self):

return self.salary

def set\_name(self,name):

self.name=name

def set\_age(self,age):

self.age=age

def set\_salary(self,salary):

self.salary=salary

class Manager(Employee):#继承了Employee类的所有方法

def \_\_init\_\_(self,name,age,salary,bouns):#重载初始化方法增加bouns属性

print('Created a class:',self.\_\_class\_\_)

self.name=name

self.age=age

self.salary=salary

self.bouns=bouns

def set\_bouns(self,bouns):

self.bouns=bouns

def getter\_bouns(self):

return self.bouns

#下面测试所有类方法是否正确

print("############# This is a test ##################\n\n")

xiaozhang=Employee('xiaozhang',20,1000)

print('get my name',xiaozhang.getter\_name())

print('get my age',xiaozhang.getter\_age())

print('get my salary',xiaozhang.getIncome())

print('set my name',xiaozhang.set\_name('person'))

print('set my age',xiaozhang.set\_age(21))

print('set my salary',xiaozhang.set\_salary(2000))

print('after set my name my name is:',xiaozhang.getter\_name())

print('after set my my age is:',xiaozhang.getter\_age())

print('after set my salary my salary is:',xiaozhang.getIncome())

print('')

laoban=Manager('laoban',50,300000,10000)

print('get my name',laoban.getter\_name())

print('get my age',laoban.getter\_age())

print('get my salary',laoban.getIncome())

print('get my bouns',laoban.getter\_bouns())

print('set my name',laoban.set\_name('boss'))

print('set my age',laoban.set\_age(51))

print('set my salary',laoban.set\_salary(20000))

print('set my bouns',laoban.set\_bouns(10000))

print('after set my name my name is:',laoban.getter\_name())

print('after set my my age is:',laoban.getter\_age())

print('after set my salary my salary is:',laoban.getIncome())

print('after set my my bouns is:',laoban.getter\_bouns())

print('')

# -\*- coding: utf-8 -\*-

class Mylist: #定义类Mylist

\_\_mylist = []

def \_\_init\_\_(self,\*args):

self.mylist = []

for arg in args:

self.\_\_mylist.append(arg)

def \_\_add\_\_(self,x): #定义操作+

for i in range(len(self.\_\_mylist)): #依次遍历列表，执行加操作

self.\_\_mylist=self.mylist+x

return self.\_\_mylist

def \_\_sub\_\_(self,x):

for i in range(0, len(self.\_\_mylist)): # 依次遍历列表，执行减操作

self.\_\_mylist[i] = self.\_\_mylist[i] - x

return self.\_\_mylist

def \_\_mul\_\_(self,x):

for i in range(0, len(self.\_\_mylist)): # 依次遍历列表，执行乘操作

self.\_\_mylist[i] = self.\_\_mylist[i] \* x

return self.\_\_mylist

def \_\_div\_\_(self,x):

for i in range(0, len(self.\_\_mylist)): # 依次遍历列表，执行除操作

self.\_\_mylist[i] = self.\_\_mylist[i] / x

return self.\_\_mylist

def \_\_mod\_\_(self,x):

for i in range(0, len(self.\_\_mylist)): # 依次遍历列表，执行求余操作

self.\_\_mylist[i] = self.\_\_mylist[i] % x

return self.\_\_mylist

def \_\_pow\_\_(self,x):

for i in range(0, len(self.\_\_mylist)): # 依次遍历列表，计算x次方

self.\_\_mylist[i] = self.\_\_mylist[i] \*\* x

return self.\_\_mylist

def show(self): #显示列表中的数值

print(self.\_\_mylist)

ll=[1,2,3,4,5,6,7,8,9,10]

list1=Mylist(ll)

print(list1.\_\_add\_\_(10))

print(list1.\_\_sub\_\_(10))

print(list1.\_\_mul\_\_(10))

print(list1.\_\_mod\_\_(10))

print(list1.\_\_pow\_\_(10))

print(list1.show())

"""

 请编写一个职工类Employee，要求其包含name、age和salary（月薪），并且完成相应getter、setter方法和获得收入getIncome的方法与构造器。职工的收入就是工资。

 请编写一个经理类Manager，要求其包含name、age和salary（月薪）、bonus（奖金），并且完成获得收入等的方法与构造器。

重载运算符

class Mylist: #定义类Mylist

\_\_mylist = []

def \_\_init\_\_(self,\*args):

self.mylist = []

for arg in args:

self.\_\_mylist.append(arg)

def \_\_add\_\_(self,x): #定义操作+

for i in range(0,len(self.\_\_mylist)): #依次遍历列表，执行加操作

self.\_\_mylist[i] = self.\_\_mylist[i]+x

return self.\_\_mylist

def \_\_sub\_\_(self,x):

pass

def \_\_mul\_\_(self,x):

pass

def \_\_div\_\_(self,x):

pass

def \_\_mod\_\_(self,x):

pass

def \_\_pow\_\_(self,x):

pass

def show(self): #显示列表中的数值

print(self.\_\_mylist)

if \_\_name\_\_ == '\_\_main\_\_': #通过name的内置属性

l = Mylist(1,2,3,4,5) #定义一个列表对象

l+10

l.show()

"""