1. 如何判断一个类是否是另一个类的子类?
2. 如何判断对象那个a是否为类A的实例对象?
3. 什么是组合?
4. 继承的特点有哪些?
5. 继承和组合有什么区别?
6.
7. 如何避免访问对象不存在的属性?
8. property函数的作用是什么?
9. 类对象是什么时候后产生的?
10. 如果对象的属性和方法名相同，会发生什么事情?
11. 当子类定义了同名的属性或方法时，Python是否会自动删除父类的相关属性或方法?
12. 假设已经有鸟类的定义（鸟类中有飞的方法），现在要定义企鹅类继承于鸟类，但是企鹅不会飞，此时应该如何屏蔽父类中的飞的方法
13. super的用法有哪些?
14. 请补充代码，让程序正常执行

```
class C:
```

```
    def __init__(self, size=10):
        self.size = size

    def getXSize(self):
        return self.size

    def setXSize(self, value):
        self.size = value

    def delXSize(self):
        del self.size

        # 此处应该补充一句代码，程序才能正
常运行

>>> c.x
10
>>> c.x = 12
>>> c.x
12
```

15. 以下类定义中，哪些是类属性，哪些是实属性?

```
class C:
        num = 0
        def __init__(self):
                self.x = 4
                self.y = 5
                C.count = 6
```

16. 以下代码中，bb对象为什么调用printBB()方法失败?

```
class BB:
        def printBB():
                print("no zuo no
die")

>>> bb = BB()
>>> bb.printBB()
Traceback (most recent call last):
  File "<pyshell#8>", line 1, in
<module>
    bb.printBB()
TypeError: printBB() takes 0
positional arguments but 1 was given
```

17. *多重继承使用不当，会导致重复调用（也称之为钻石继承、菱形继承）的问题，请分析一下代码在实际编程中，有可能导致什么问题?

```
class A():
    def __init__(self):
        print("进入A...")
        print("离开A...")

class B(A):
    def __init__(self):
        print("进入B...")
        A.__init__(self)
        print("离开B...")

class C(A):
    def __init__(self):
        print("进入C...")
        A.__init__(self)
        print("离开C...")

class D(B, C):
    def __init__(self):
        print("进入D...")
        B.__init__(self)
        C.__init__(self)
```

```
        print("离开D...")
```

18. *如何解决上一题出现的问题?

19. *利用修饰器修改以下代码

```
class CodeA:
    def fun():
        print("调用静态方法 foo()")

        # 将 foo() 方法设置为静态方法
        fun = staticmethod(foo)
```

```
class CodeB:
    def fun(cls):
        print("调用类方法 foo()")

        # 将 foo() 方法设置为类方法
        fun = classmethod(foo)
```

20. *请将以下代码修改为没有修饰器的等同形式

```
@something
def f():
    print("I love FishC.com!")
```

21. *将第5题的代码修改为'使用属性修饰器创建描述符'的方式实现?

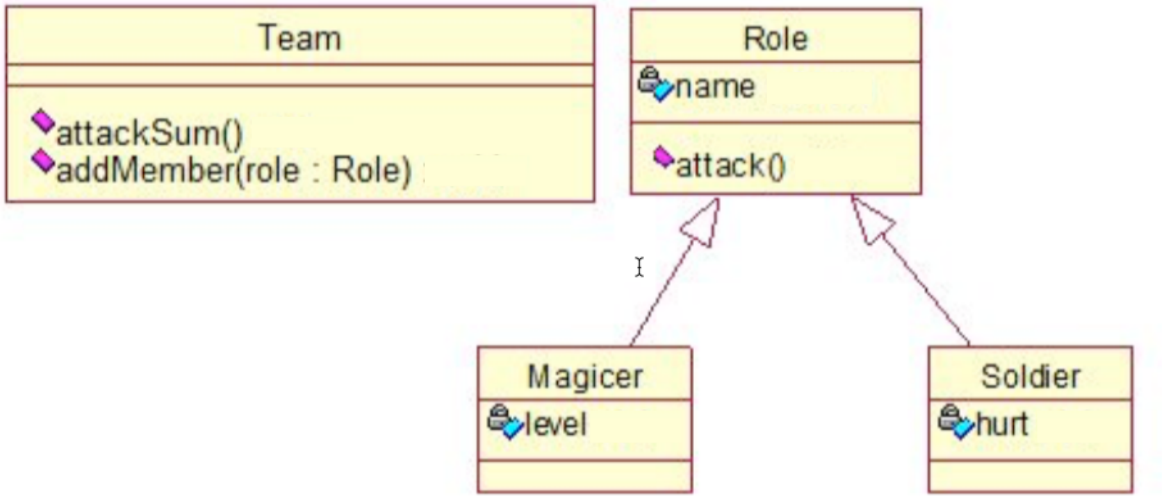
22. **创建三个类，组成一个继承关系，表示游戏中的角色

    描述如下:
    父类：Role，是所有职业的父类
    属性：name，表示角色的名字
    方法：attack()，返回值为角色的攻击对敌人的伤害

    Role有两个子类:
    1. 魔法师：Magicer
        属性：魔法等级(范围：1~10)
        方法：attack()，该方法返回法师的攻击对敌人造成的伤害值
        法师攻击伤害值为:魔法等级*魔法基本伤害值(固定为5)
    2. 战士Soldier
        属性:攻击伤害值
        方法:attack()，该方法返回战士的攻击对敌人造成的伤害值。
        战士的攻击伤害值为:其攻击伤害属性值
    3. 再设计一个Team 类，表示一个组队。具有如下方法

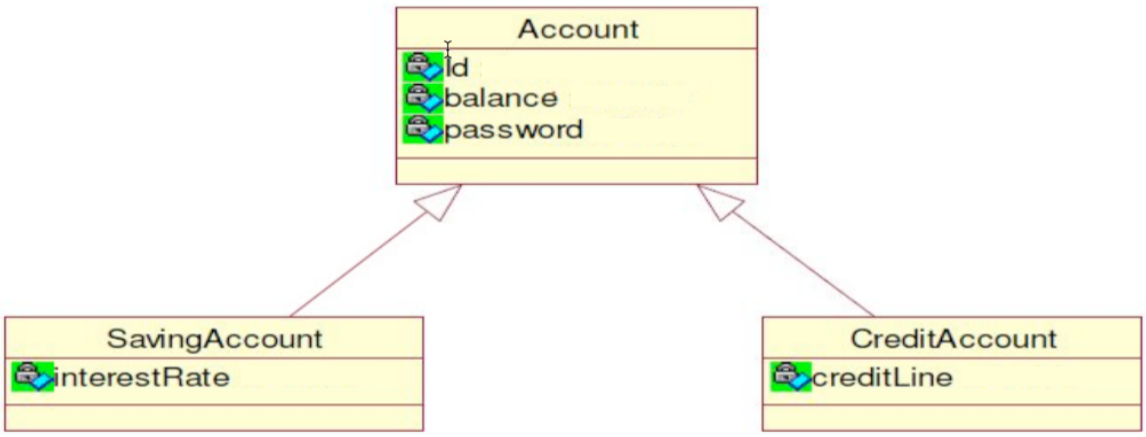
1） addMember，表示组队增加一个成员。注意:组队成员最多为6 人 提示:应当利用一个数组属性，保存所有成员

2） attackSum，表示组队所有成员进行攻击时，对敌人造成的总伤害值



请根据类图和描述，创建相应的类，并编写相应的测试代码

23. **设计如下继承关系：



Accout 表示银行账户

id 属性表示账户id

balance 表示账户余额

password 表示账户密码

SavingAccount 表示储蓄账户

interestRate 表示存款利率

CreditAccount 表示信用账户

creditLine 表示信用额度。

完成下列任务：

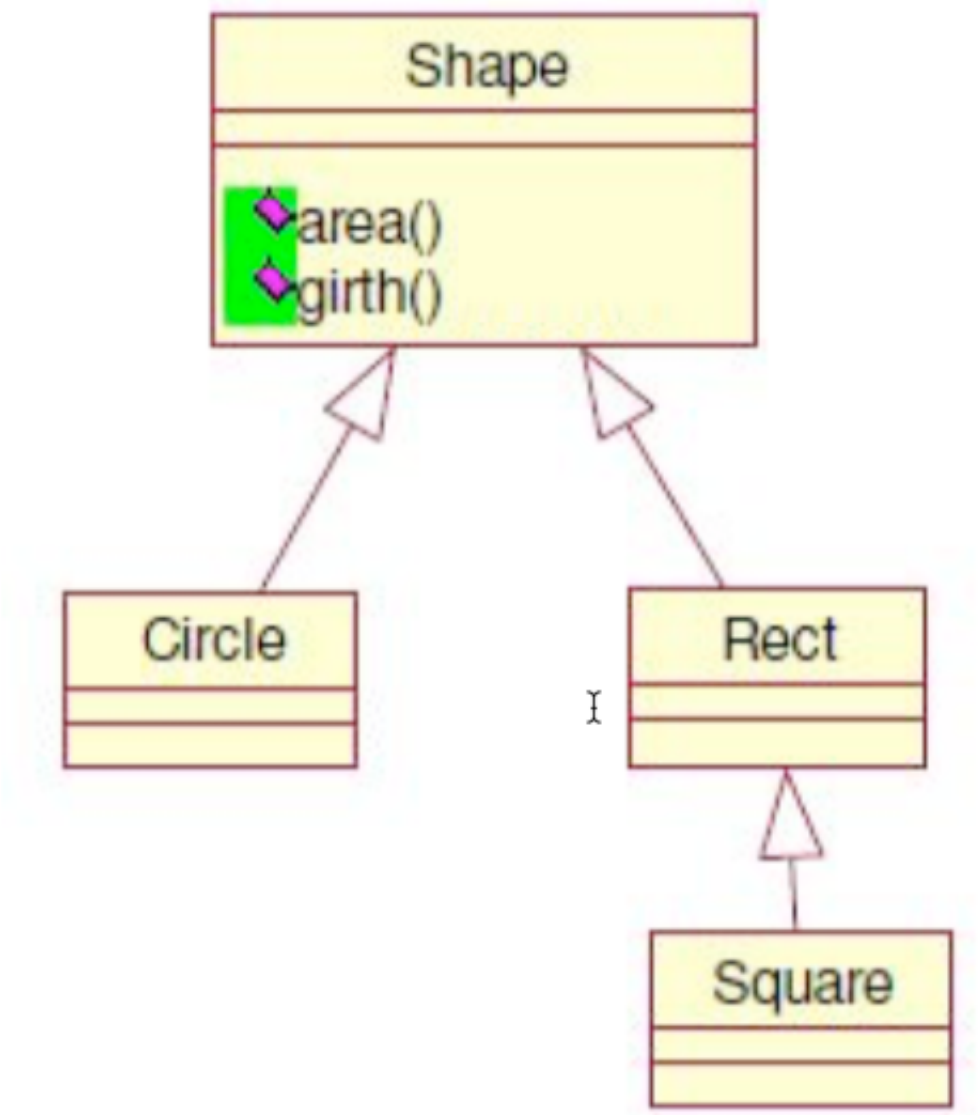
1) 修改setPassword 方法，要求：

setPassword 判断新密码长度是否是6 位，如果不是则不予修改；

修改getPassword 方法，要求每次都返回None值。

2) 修改 interestRate 的 set 方法，要求利率大于 0 并小于 10%。

24. **完成以下代码：



a) Circle 类(圆形)，属性:半径;方法:求周长、求面积

b) Rect 类(矩形)，属性:长、宽;方法:求周长、求面积

c) Square 类(正方形)，属性:边长;方法:求周长、求面积

提示:

1) 这三个类均具有求周长和面积的方法

2) 正方形是特殊的矩形

25. 某公司的雇员分为以下若干类:

Employee:这是所有员工总的父类，属性:员工的姓名,员工的生日月份。方法:getSalary(intmonth) 根据参数月份来确定工资，如果该月员工过生日，则公司会额外奖励 100 元。

SalariedEmployee:Employee 的子类，拿固定工资的员工。属性:月薪

HourlyEmployee:Employee 的子类，按小时拿工资的员工，每月工作超出160 小时的部分按照 1.5 倍工资发放。属性:每小时的工资、每月工作的小时数

SalesEmployee:Employee 的子类，销售人员，工资由月销售额和提成率决定。

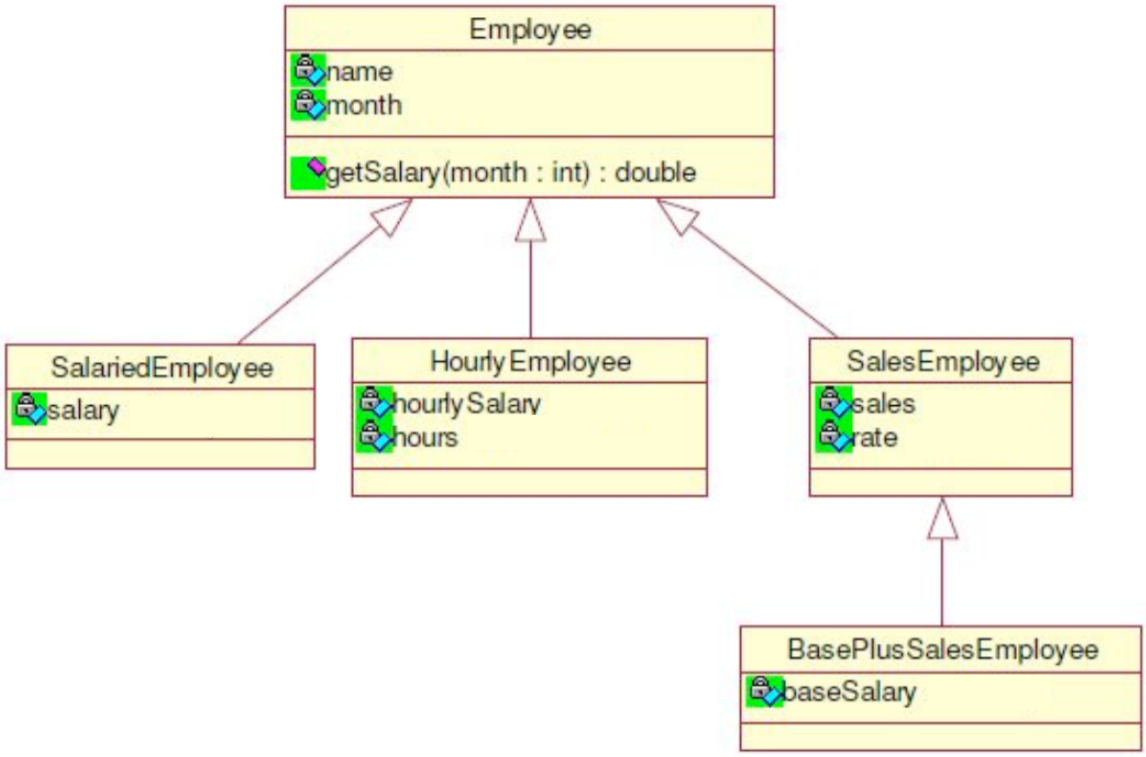
属性:月销售额、提成率

BasePlusSalesEmployee:SalesEmployee 的子类，有固定底薪的销售人员，工资由底薪加上销售提成部分。属性:底薪。根据要求创建

SalariedEmployee、HourlyEmployees、SaleEmployee 和

BasePlusSalesEmployee四个类的对象各一个，并计算某个月这四个对象的工资。注意:要求把每个类都做成完全封装，不允许非私有化属性。

类图如下:



26. *在上一题的基础上，创建一个Employee 列表，分别创建若干不同 的Employee对象，并打印某个月的工资。

27. **请定义一个栈（Stack），用于模拟一种后进先出（LIFO）或者先进后出（FILO）的数据结构，至少要有一下方法

方法名	含义
isEmpty()	判断当前栈是否为空，为空则返回True否则返回False
push()	向栈顶部压入一个数据（存储到栈中）
pop()	从栈顶弹出一个数据（并从栈中删除）
top()	显示当前栈顶的一个数据
bottom()	显示当期那栈底的一个数据

28. ***在第23题的基础上，创建一个Bank 类，其中包括三个方法:开户、存款、取款

a) 开户：
openAccount(id, password,type)
其中，id 表示账户id，password 表示账户密码，type 表示账户类型。如果type 为0则创建一个Account 账户，如果type 为1 则创建一个储蓄账 户SavingAccount，如果type为2 则创建一个信用账户CreditAccount。返回值为开户时创建的Account 对象
b)存款：
deposit(a,amount)
其中，a 表示存入账号，amount 表示存入的金额。返回值表示存款之后 的余额
c) 取款
withdraw(a,amount)
其中，a 表示取款账号，amount 表示取出的金额，返回值表示取款之后的余额。特别的，除非Account 类型是CreditAccount，否则不允许透支。

29. ***定义一个点类（Point）和直线类（Line），使用getLen方法可以获得直线的长度。

    提示：

    1. 设点$A(X_1, Y_1)$,点$B(X_2, Y_2)$，则两点构成的直线长度为：

       |AB|=$\sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$

2. Python中计算开根号可以使用math模块中的sqrt函数
3. 直线需要有两点构成，因此初始化时需要有两个点对象（Point）作为参数