Python中的类的继承

类的继承

1.继承

- python中的类支持继承，并且支持多继承。
- python中默认情况是继承自object（object是python中所有类的基类）

a.什么是继承

- 一个类可以继承另外一个类，继承者我们叫子类，被继承者叫父类。继承就是让子类直接拥有父类中的内容

b.可以继承哪些内容

- 所有的属性和方法都可以继承

```
class Person(object):
    num = 61

    def __init__(self):
        self.name = '张三'
        self.age = 0
        self.sex = '男'

    def show_message(self):
        print('%s你好吗?' % self.name)

# Student类继承自Person类
class Student(Person):
    pass

# 创建学生对象
stu1 = Student()
# 对象属性可以继承
print(stu1.name, stu1.age, stu1.sex)

# 类的字段可以继承
print(Student.num)

# 对象方法可以继承
stu1.show_message()
```

子类–添加方法

- 子类除了拥有从父类继承下来的属性和方法，还拥有属于自己的属性和方法

1.在子类中添加方法

a.添加一个新的方法

- 直接在子类中声明其他的方法;
  添加后子类可以调用自己的方法也可以调用父类的方法，但是父类不能调用子类的方法

b.重写父类的方法: 重新实现父类的方法

完全重写 – 覆盖父类的功能 – 直接在子类中重新实现父类的方法
部分重写 – 保留父类的功能，添加新的功能 – 在子类中实现父类方法的时候通过super()去调用父类的方法，再添加新的功能
注意:
a.可以子类的方法中通过super()去调用父类的方法
super(类, 对象)– 获取对象中父类的部分(要求对象是这个指定的类的对象)
b.静态方法中不能使用super()
c.类中方法的调用过程
通过对象或者类调用方法的时候，先看当前类中是否声明过这个方法，如果声明过就直接调用当前类对应的方法;
如果当前类中没有声明过，会去找父类中有没有声明过这个方法，声明过就调用父类的方法;
如果父类中也没有声明过，就去找父类的父类...以此类推，直到object中也没有声明过，程序才会崩溃

```
class Person:
    # 类的字段
    num = 61

    # 对象属性
    def __init__(self):
        self.name = '张三'
        self.age = 0
        self.sex = '男'

    def fun1(self):
        print('Person的对象方法')

    # 方法
    def show_message(self):
        print('%s,你好吗？' % self.name)


class Student(Person):

    def study(self):
        print('%s在学生' % self.name)

    # 保留父类的功能
    def show_message(self):
```

```
        super().show_message()
        print('我去上学~')
        super().fun1()

# Student.info()
```

子类–添加属性

1.添加类的字段

- 直接在子类中添加新的字段

2.添加对象属性

- 继承对象属性是通过继承父类的init方法继承下来的
- 如果想要在保留父类继承下来的对象属性的前提下，添加新的对象属性，
  需要在子类的init方法中，通过super()去调用父类的init方法

```
class Person:
    num = 61

    def __init__(self, name):
        self.name = name
        self.age = 0

class Student(Person):
    number = 100

    def __init__(self, name):
        super().__init__(name)
        self.study_id = '001'

print(Student.number, Student.num)

stu1 = Student('小明')
print(stu1.name, stu1.age, stu1.study_id)

# 练习：
# 声明一个动物类，有属性：年龄，颜色，类型。
#  要求创建动物对象的时候类型和颜色必须赋值，年龄可以赋值也可以不赋值
# 声明一个猫类，有属性:年龄，颜色，类型，爱好
# 要求创建猫对象的时候，颜色必须赋值，年龄和爱好可以赋值也可以不赋值，类型不能赋值

class Aniaml:
    def __init__(self, type, color, age=0):
        self.type = type
        self.color = color
        self.age = age
```

```
class Cat(Aniaml):
    def __init__(self, color, age=0, hobby=''):
        super().__init__('猫科', color, age)
        self.hobby = hobby


an1 = Aniaml('犬科', '黄色')
an2 = Aniaml('犬科', '黄色', 10)

cat1 = Cat('白色')
cat2 = Cat('灰色', 3)
cat3 = Cat('灰色', hobby='睡觉')
cat4 = Cat('灰色', 3, '睡觉')
```

多继承

多继承就是让一个类同时继承多个类

- 注意：实际开发一般不使用多继承

```
class Animal:
    num=61
    def __init__(self,name='小红',age=0,color='黑色'):
        self.name = name
        self.age = age
        self.color = color
    def show_info(self):
        print('名字：%s 年龄：%s 颜色：%s'%(self.name,self.age,self.color))

class Fly:
    info = '飞'
    def __init__(self,distance=0,speed=0):
        self.distance = distance
        self.speed = speed
    @staticmethod
    def show():
        print('飞')
#让birds同时继承Animal,Fly
class Birds(Animal,Fly):
    pass
#两个类的字段都会继承
print(Birds.num,Birds.info)
Birds.show()

#两个类的方法都能继承，对象属性只能继承第一个类的对象属性
b1 = Birds()
print(b1.name)
```