任务9-1 面向对象基本概念

面向对象

程序员“面向对象”

在现实世界中存在各种不同形态的事物，这些事物之间存在着各种各样的联系。在程序中使用对象来映射现实中的事物，使用对象间的关系来描述事物之间的联系，这种思想就是面向对象。

面向对象

分别使用面向过程和面向对象来实现五子棋

编程思想	实现步骤	特点
面向过程	✓ 开始游戏	先分析解决问题的步骤 使用函数把这些步骤以此实现 使用的时候需要逐个调用函数
	✓ 黑子先走	
	✓ 绘制画面	
	✓ 轮到白子。	
	✓ 绘制画面	
	✓ 判断输赢	
	✓ 返回步骤2	
	✓ 输出最后结果	
面向对象	✓ 黑白双方：这两方的行为一样	把解决问题的事物分为多个对象 对象具备解决问题过程中的行为
	✓ 棋盘系统：负责绘制画面	
	✓ 规则系统：负责判断诸如犯规、输赢等。	

面向对象

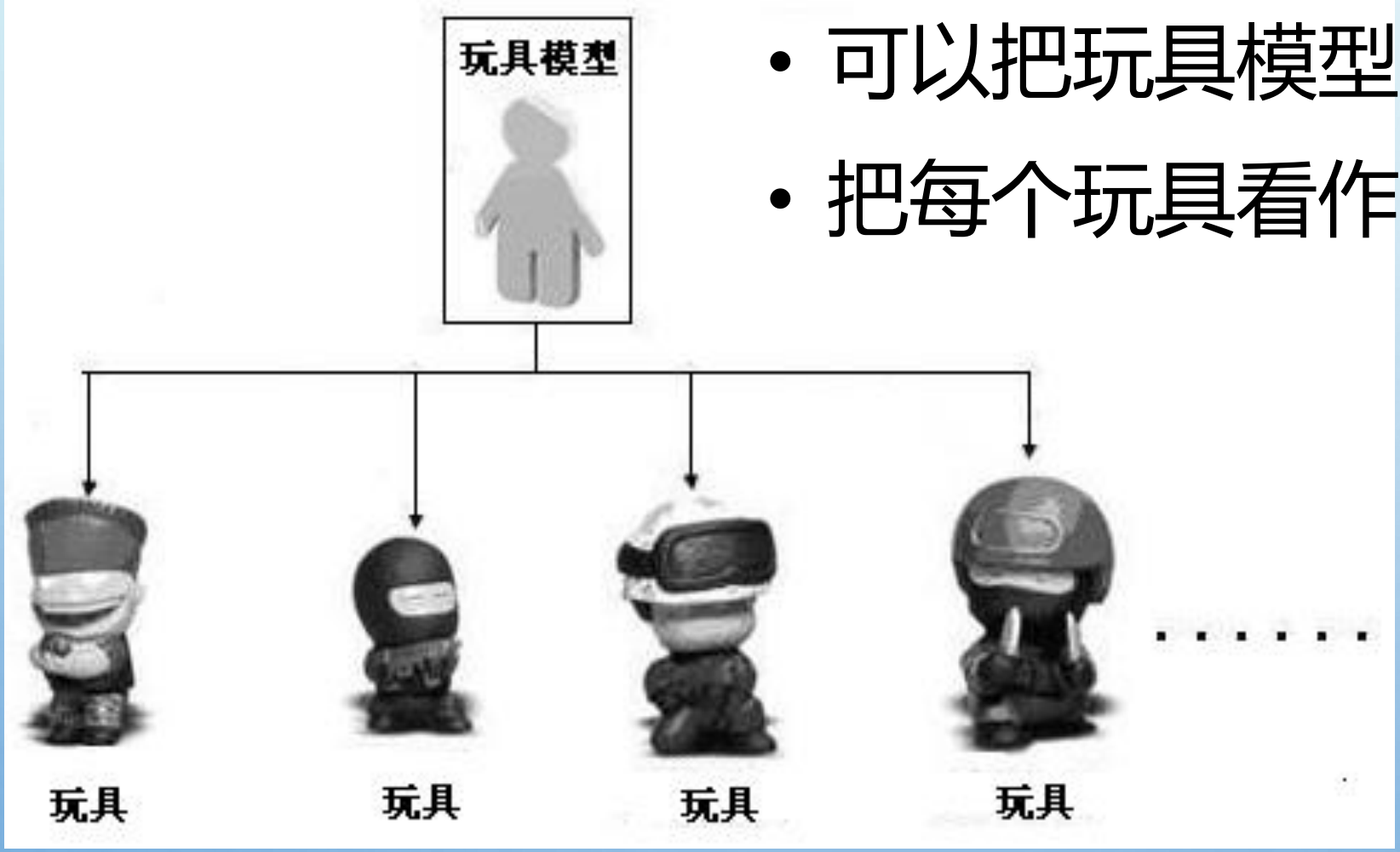
若加入悔棋功能，面向过程和面向对象，分别怎么实现呢?

面向过程	面向对象
从输入到判断到显示的一系列步骤都需要改动	只需要改动棋盘对象就可以 更简便！

类和对象的关系

- 面向对象编程有两个非常重要的概念：类和对象。
- 对象是面向对象编程的核心。
- 具有相似特征和行为的事物的集合统称为类
- 对象是根据类创建的，一个类可以对应多个对象。

类和对象的关系



- 可以把玩具模型看作一个类
- 把每个玩具看作一个对象

类的定义

类是由3部分组成的：

- 类的名称：类名，首字母必须大写，比如Person。
- 类的属性：一组数据，比如性别。
- 类的方法：允许进行操作的方法，比如说话。

类的定义

使用class关键字来声明一个类，基本格式如下：

class 类名:

类的属性

类的方法

根据类创建对象

根据类创建对象的语法格式如下：

```
对象名 = 类名()
```

要想给对象添加属性，可以通过如下方式：

```
对象名.新的属性名 = 值
```

构造方法

- 构造方法指的是__init__方法。
- 当创建类的实例的时候，系统会自动调用构造方法，从而实现对类进行初始化的操作。

构造方法

```
bmw = Car("雪山白")
```

```
class Car:
    def __init__(self):
        self.color = “黑色”
    def toot(self):
        print("%s的车在鸣笛..."%(self.color))
```

析构方法

- 当删除一个对象来释放类所占用资源的时候，Python解释器默认会调用另外一个方法，这个方法就是__del__( )方法。
- __del__方法被称为析构方法。

self的使用

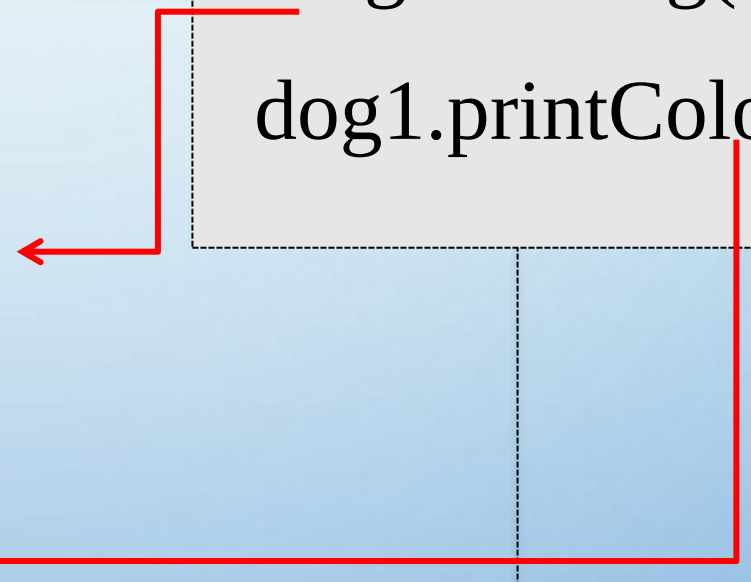
- 在方法的列表中，第1个参数永远都是self。
- self的字面意思是自己，我们可以把它当做C++里面的this指针理解，表示的是对象自身。
- 当某个对象调用方法的时候，Python解释器会把这个对象作为第1个参数传给self，开发者只需要传递后面的参数就可以了。

self的使用

示例：

```
class Dog:
    def __init__(self, newColor):
        self.color = newColor
    def printColor(self):
        print("颜色为： %s"%self.color)
```

```
dog1 =  Dog("白色")
dog1.printColor()
```



运算符重载

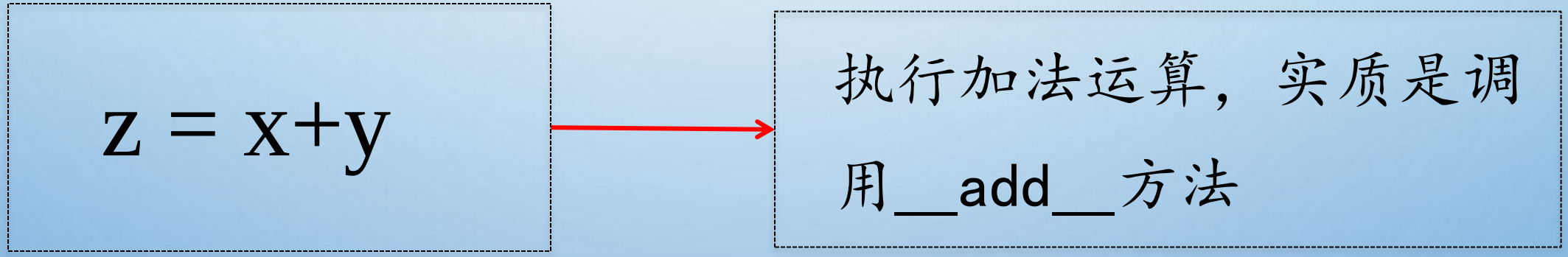
- 运算符重载是通过实现特定的方法使类的实例对象支持Python的各种内置操作 。例如： +运算符是类里提供的__add__这个函数，当调用+实现加法运算的时候，实际上是调用了__add__方法。

运算符重载

方法	说明	何时调用方法
__add__	加法运算	对象加法：x+y，x+=y
__sub__	减法运算	对象减法：x-y，x-=y
__mul__	乘法运算	对象乘法：x*y，x*=y
__diy__	除法运算	对象除法：x/y，x/=y
__getitem__	索引，分片	x[i]、x[i:j]、没有__iter__的for循环等
__setitem__	索引赋值	x[i]=值、x[i:j]=序列对象
__delitem__	索引和分片删除	del x[i]、del x[i:j]

加法运算符重载

加法运算是通过调用__add__方法完成重载的，当两个实例对象执行加法运算时，自动调用__add__方法。



索引和分片重载

跟索引相关的重载方法包括如下3个：

- __getitem__：索引、分片；
- __setitem__：索引赋值；
- __delitem__：索引和分片删除。

索引和分片重载

1. __getitem__方法

在对实例对象执行索引、分片或者for迭代操作时，会自动调用__getitem__方法。

```
# 定义索引、分片运算符重载方法
    def __getitem__(self, index):
        return self.data[index]
```

索引和分片重载

2. __setitem__方法

通过赋值语句给索引或者分片赋值时，调用__ setitem __方法实现对序列对象的修改。

```
def __setitem__(self, index, value):
        self.data[index] = value
```

索引和分片重载

3. __delitem__方法

当调用del方法时，实质上会调用__delitem__方法实现删除操作。

```
def __delitem__(self, index):
        del self.data[index]
```