

实验六 面向对象编程

一、实验目的

1. 掌握定义类的方法
2. 掌握创建和使用对象的方法
3. 掌握类的继承的概念和使用方法

二、实验内容

1、上课笔记（拍照、签名）

2、验证以下内容：

- 1) 类的定义和使用 例 10.2~4
- 2) 类的方法 例 10.7
- 3) 类的属性 例 10.8~10
- 4 类的继承 例 10.11~13

3、编程完成以下功能

1. 定义一个 Dog 类，类属性有名字 name、毛色 color、体重 weight，方法为有 bark，调用该方法时返回“wang! wang!”，show，调用该方法时输出“一只体重为 weight 的 color name 在 wang! wang! 叫”。使用 Dog 类创建一个对象，名字为“旺财”，毛色为“黄色”，体重为“10”，并调用 show 方法。

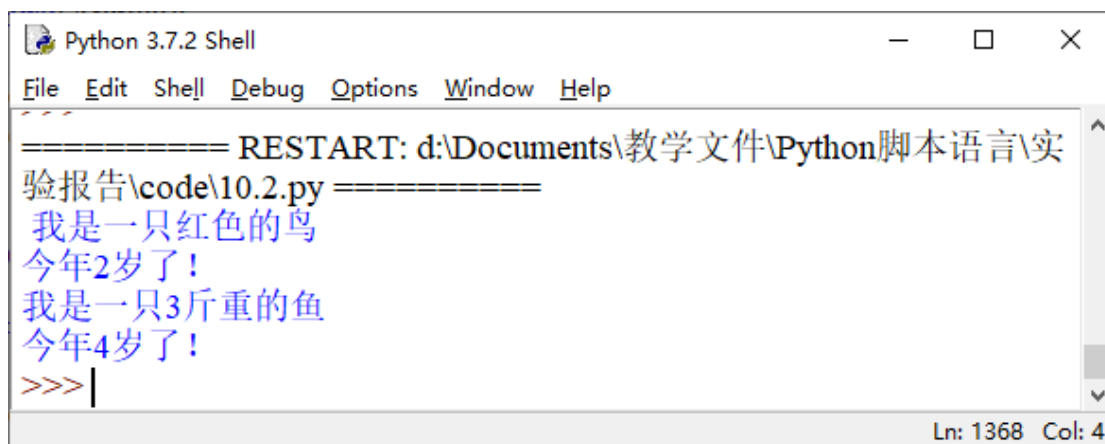
```
class Dog:
```

```
    dog=Dog('旺财','黄色',10)
```

```
    dog.show()
```

```
===== RESTART: d:/Documents/教学文件/Python脚本语  
=====  
一只重10的黄色 旺财在wang! wang! 叫  
>>> |
```

2. 设计 Bird 类和 Fish 类，都继承自 Animal 类，实现方法 show(), 输出信息。参与输出结果如下图。



```
Python 3.7.2 Shell  
File Edit Shell Debug Options Window Help  
===== RESTART: d:\Documents\教学文件\Python脚本语言\实  
验报告\code\10.2.py =====  
我是一只红色的鸟  
今年2岁了!  
我是一只3斤重的鱼  
今年4岁了!  
>>> |  
Ln: 1368 Col: 4
```

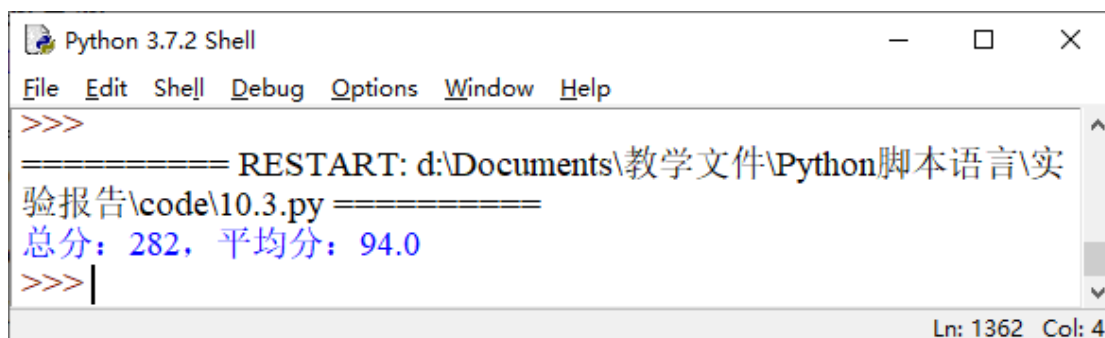
3. 设计一个 Student 类，在类中定义两个方法，一个方法 input 用来输入某个学生的 3 门成绩，另一个方法 output 计算学生的总分和平均分并输出。

```
class Student:
```

```
s1 = Student()
```

```
s1.input(93,94,95)
```

```
s1.output()
```



```
Python 3.7.2 Shell  
File Edit Shell Debug Options Window Help  
>>>  
===== RESTART: d:\Documents\教学文件\Python脚本语言\实  
验报告\code\10.3.py =====  
总分: 282, 平均分: 94.0  
>>> |  
Ln: 1362 Col: 4
```

4. 王者荣耀是一款非常流行的即时对战类游戏，里面有非常多的游戏角色可供选择。所有的角色都具有以下操作：普通攻击，技能攻击。

创建两个英雄角色：

- 关羽：普通攻击 10，技能攻击“单刀赴会”。
- 吕布：普通攻击 15，技能攻击“贪狼之握”，使用时播放“谁敢战我”。

创建一个控制类，能够操作英雄角色使用普通攻击或技能攻击。运行效果如图所示。



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
===== RESTART: d:\Documents\教学文件\Python脚本语言\实
验报告\code\10.4.py =====
关羽使用了普通攻击，攻击力为10
关羽使用了技能攻击单刀赴会
吕布使用了普通攻击，攻击力为15
吕布：谁敢与战！！
吕布使用了技能攻击贪狼之握
>>>
Ln: 1359 Col: 4
```

5. 设计并测试一个表示一个点的 `MyPoint` 类。

要求：该类包括以下属性：

`x`：点的横坐标。

`y`：点的纵坐标。

包括如下方法：

`__init__(self, x, y)`：构造方法，创建对象的同时为属性 `x`、`y` 赋初值。

`getX()`：获得点的横坐标。

`getY()`：获得点的纵坐标。

`getDdistance(self, p)`：返回当前点与点 `p` 之间的距离。

6. 按照以下要求定义一个乌龟类和鱼类并尝试编写游戏。

假设游戏场景范围为 (x, y) , $0 \leq x \leq 10, 0 \leq y \leq 10$ 。

- (1) 游戏生成 1 只乌龟和 10 条鱼;
- (2) 它们的移动方向均随机;
- (3) 乌龟的最大移动能力是 2 (可以随机移动 1 或 2), 鱼的最大移动能力是 1, 移动是沿 x 轴或 y 轴进行;
- (4) 当移动到场景边界, 移动向反方向移动;
- (5) 乌龟初始化体力为 100;
- (6) 乌龟每移动一次, 体力消耗 1;
- (7) 当乌龟和鱼坐标重叠, 乌龟吃掉鱼, 乌龟体力增加 20;
- (8) 鱼暂不计算体力;
- (9) 当乌龟体力值为 0 或者鱼儿的数量为 0, 游戏结束

分析:

根据上述要求, 此游戏有两类对象, 分别是乌龟和鱼, 它们分别具有的能力如下。

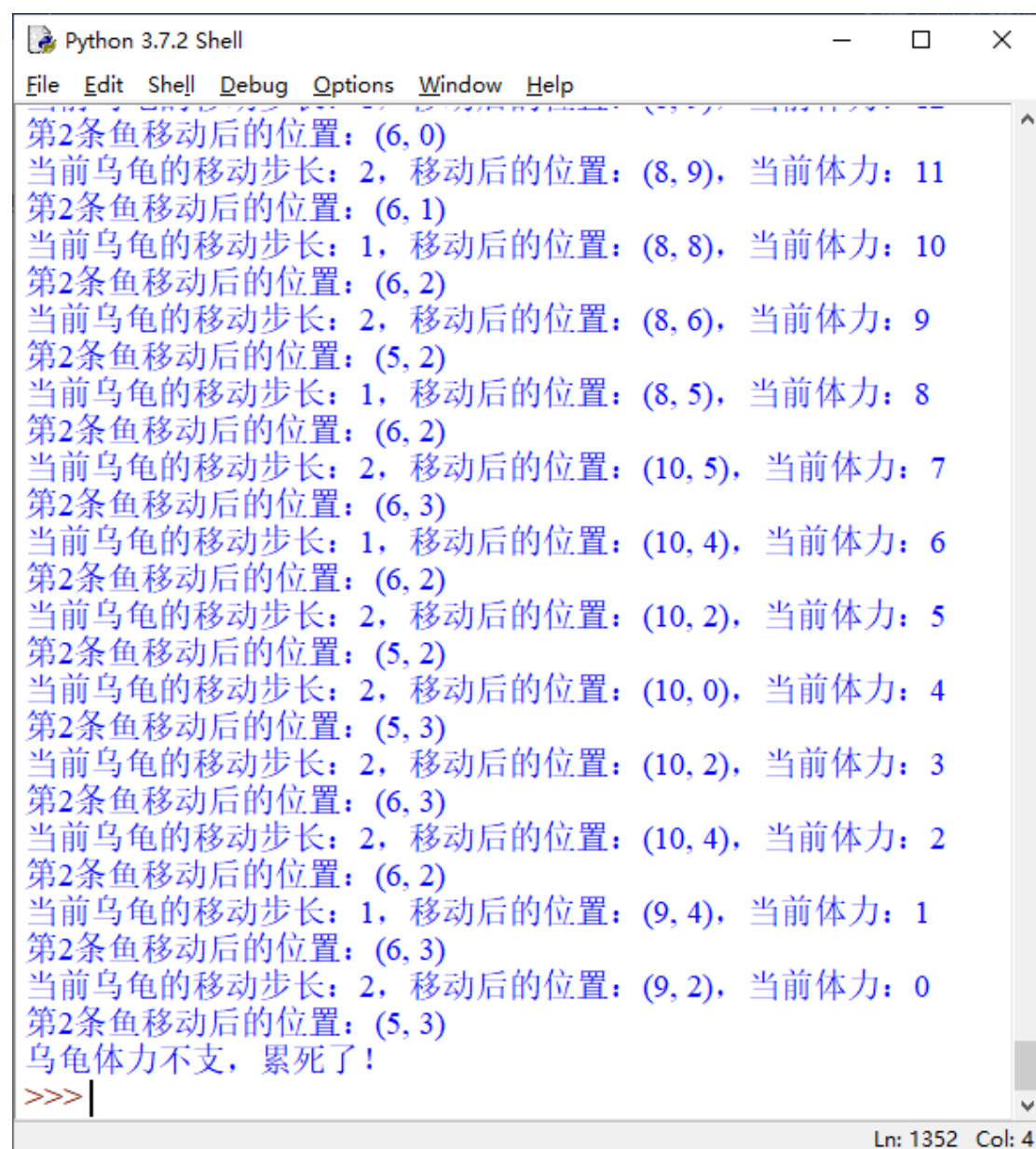
(1) 乌龟: 初始化位置 (随机), 移动 (随机 1 或 2, 遇到边界向反方向移动), 吃鱼 (乌龟坐标和鱼坐标重叠, 乌龟吃鱼);

(2) 鱼: 初始化位置 (随机), 移动 (移动 1, 遇到边界向反方向移动)。

以上分析表明, 乌龟和鱼类都有“随机初始化位置”和“移动”的方法, 为此抽象出它们的父类 `GameAnimal`, 在父类中定义初始化位置(`initLocation`)和移动(`move`)的方法。此游戏包括三个类, `GameAnimal`, 乌龟(`Tortoise`)和鱼(`Fish`)。

在主程序中, 定义 1 个乌龟对象和 10 个鱼对象, 然后循环执行如下步骤:

- (1) 获取乌龟的坐标;
- (2) 判断所有鱼当中是否有鱼的坐标与乌龟的坐标重叠, 如果重叠则删除此鱼对象, 乌龟体力增加 20;
- (3) 若鱼被吃完, 退出循环;
- (4) 若乌龟能量为 0, 退出循环。



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
第2条鱼移动后的位置: (6, 0)
当前乌龟的移动步长: 2, 移动后的位置: (8, 9), 当前体力: 11
第2条鱼移动后的位置: (6, 1)
当前乌龟的移动步长: 1, 移动后的位置: (8, 8), 当前体力: 10
第2条鱼移动后的位置: (6, 2)
当前乌龟的移动步长: 2, 移动后的位置: (8, 6), 当前体力: 9
第2条鱼移动后的位置: (5, 2)
当前乌龟的移动步长: 1, 移动后的位置: (8, 5), 当前体力: 8
第2条鱼移动后的位置: (6, 2)
当前乌龟的移动步长: 2, 移动后的位置: (10, 5), 当前体力: 7
第2条鱼移动后的位置: (6, 3)
当前乌龟的移动步长: 1, 移动后的位置: (10, 4), 当前体力: 6
第2条鱼移动后的位置: (6, 2)
当前乌龟的移动步长: 2, 移动后的位置: (10, 2), 当前体力: 5
第2条鱼移动后的位置: (5, 2)
当前乌龟的移动步长: 2, 移动后的位置: (10, 0), 当前体力: 4
第2条鱼移动后的位置: (5, 3)
当前乌龟的移动步长: 2, 移动后的位置: (10, 2), 当前体力: 3
第2条鱼移动后的位置: (6, 3)
当前乌龟的移动步长: 2, 移动后的位置: (10, 4), 当前体力: 2
第2条鱼移动后的位置: (6, 2)
当前乌龟的移动步长: 1, 移动后的位置: (9, 4), 当前体力: 1
第2条鱼移动后的位置: (6, 3)
当前乌龟的移动步长: 2, 移动后的位置: (9, 2), 当前体力: 0
第2条鱼移动后的位置: (5, 3)
乌龟体力不支, 累死了!
>>> |
```

Ln: 1352 Col: 4