第1题

```
#1题 定义Rugby为Athlete的子类，并增加子类自己的属性squat。（5分）

class Athlete:
    def __init__(self,a_name,a_dob=None,a_times=[]):
        self.name = a_name
        self.dob = a_dob
        self.times = a_times
    def top3(self):
        return sorted(set([self.sanitize(t) for t in self.times]))[0:3]
    def sanitize(self,time_string):
        if '-' in time_string:
            splitter = '-'
        elif ':' in time_string:
            splitter = ':'
        else:
            return (time_string)
        (mins,secs) = time_string.split(splitter)
        return (mins+'.'+secs)



class Rugby(Athlete):
#代码1，定义Rugby类继承Athlete
    def __init__(self,a_name,a_dob,a_squat,a_times):#新局部变量squat
        Athlete.__init__(self,a_name,a_dob,a_times)
        #代码2，调用父类的构造方法，传递的参数为a_dob、a_times
        self.squat=a_squat
        #代码3，将a_squat赋值给类属性squat
```

第2题

```
#2题 定义OtherAthlete类为Athlete类的子类，重写top3方法（允许重复的时间） （5分）
class OtherAthlete(Athlete):
#代码1，定义OtherAthlete类继承Athlete
    def __init__(self,a_name,a_bod,a_squat,a_times):
        Athlete.__init__(self,a_name,a_bod,a_times)
        self.squat=a_squat
    def top3(self):
        return sorted([self.sanitize(t) for t in self.times])[0:3]
    #代码2，定义无参数top3函数，对self.times属性应用统一化和排序功能
```

第3题

```
#3题 定义print_rugby函数，以多态的方式调用子类属性和方法  （5分）
def get_coach_data(filename):
    with open(filename) as f:
        line = f.readline()
    return line.strip().split(',')
loren = get_coach_data('mywork/loren.txt')
mark = get_coach_data('mywork/mark.txt')

loren = Rugby(loren.pop(0),loren.pop(0),loren.pop(0),loren)
mark = OtherAthlete(mark.pop(0),mark.pop(0),mark.pop(0),mark)
```

```
def print_rugby(athlete):
    print(athlete.name)
    print(athlete.dob)
    print(athlete.squat)
    print(athlete.top3())
    #代码1，打印athlete的属性dob、squat和top3方法的返回值

print_rugby(loren)
#代码2，调用print_rugby函数，参数为loren
print_rugby(mark)
#代码3，调用print_rugby函数，参数为mark
```

```
2011-11-3
270
3.59
['3.11', '3.23', '4.10']
mark
2010-2-4
300
['3.11', '3.11', '3.23']
```

第4题

```
#4题 有两个父类，一个Father，一个Mother，定义Child类共同继承这两个父类，子类调用父类的属性和方法  （5分）

class Father():
    def __init__(self):
        self.color = 'black'
    def talk(self):
        print("---爸爸的表达能力---")

class Mother():
    def __init__(self):
        self.height = 170
    def smart(self):
        print("---妈妈聪明的头脑---")

#代码1，定义Child类继承Father和Mother
class Child(Father,Mother):
    def __init__(self):
        #代码2，调用Mother类的的__init__方法
        Father.__init__(self)
        Mother.__init__(self)
#代码3，创建Child类的对象child,调用无参数的构造方法
child1=Child()
child1.smart()
print(child1.color)
print(child1.height)
#代码4，通过child调用父类的smart方法
#代码5，通过child打印父类的color属性
```

```
---妈妈聪明的头脑---
black
170
```