第1题 答案在第五视频前30分钟

'james,2006-11-11,2-34,3:21,2.34,2.45,3.01,2:01,2:01,3:10,2-22'

存储以上的数据，如何定义运动员类，补全代码（4分）

```
class Athlete:
    def __init__(self,a_name,a_dob=None,a_time=[]):
        self.name=a_name
        self.dob=a_dob
        self.time=a_time

    def top3(self):
      return sorted(set([self.sanitize(t) for t in self.time]))[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)
```

第2题

数据所在文件的路径为'work/james_new.txt'，使用运动员对象，补全代码（4分）

```
def get_coach_data(filename):
    with open(filename) as f:
        line = f.readline()
    return line.strip().split(',')

james_new=get_coach_data('work/james_new.txt') #从文件中读取数据
james_name=james_new.pop(0)
james_dob=james_new.pop(0)
james_time=james_new

#创建Athlete对象，将2个变量传递个构造方法，赋值给james
james=Athlete(james_name,james_dob,james_time)

print('姓名：%s,生日：%s,最快的3次成绩：%s' %(james.name,james.dob,james.time))
```

```
姓名：james,生日：2006-11-11,最快的3次成绩：['2-34', '3:21', '2.34', '2.45', '3.01', '2:01', '2:01', '3:10', '2
```

第3题

类属性，类方法，补全代码（4分）

```
class Athlete:

    #运动员集训了，要买东西的同学要把地址改一下
    address='旧基地'

    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)
    @classmethod
    def changeaddress(self):
        self.address='新基地'



james_new = get_coach_data('work/james_new.txt')
james_name = james_new.pop(0)
james_dob = james_new.pop(0)
james_times = james_new

james = Athlete(james_name,james_dob,james_times)  #创建对象
#Athlete.address='new address'  #类属性的方法
Athlete.changeaddress()    #类方法的方法
print(james.address)
print(Athlete.address)
```

```
新基地
新基地
```

第4题

将第3题中的实例变量name改为私有的属性，将sanitize改为私有方法，补全代码（4分）

```
class Athlete:
    def __init__(self,a_name,a_dob=None,a_time=[]):
        self.name=a_name
        self.dob = a_dob
        self.time = a_time

    def sayName(self):
        print(self.__name)


    def top3(self):
        return sorted(set([self.__sanitize(t) for t in self.time]))[0:3]


    #代码4
    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)
```

第5题

数据内容james,2006-11-11,2-34,3:21,2.34,2.45,3.01,2:01,2:01,3:10,2-22，以分钟.秒的形式打印'2-34'后面的所有时间。

输出的结果为'2.34', '3.21', '2.34', '2.45', '3.01', '2.01', '2.01', '3.10', '2.22',补全代码。(4分)

```
data = 'james,2006-11-11,2-34,3:21,2.34,2.45,3.01,2:01,2:01,3:10,2-22'

def sanitize(time_string):
    #判断每个时间，包含-和：就赋值个splitter，不满足则返回，对string切分
    if '-'in time_string:
        splitter='-'
    elif ':' in time_string:
        splitter=':'
    else:
        return (time_string)
    (mins,secs)=time_string.split(splitter)
    return (mins+'.'+secs)
#对data以逗号切分给James
james=data.split(',')

name = james.pop(0)
dob = james.pop(0)

#使用列表生成式对每个时间标准化后赋给times
times=[sanitize(t) for t in james]
print(times)
```

```
['2.34', '3.21', '2.34', '2.45', '3.01', '2.01', '2.01', '3.10', '2.22']
```

请点击此处查看本环境基本用法.
Please click here for more detailed instructions.