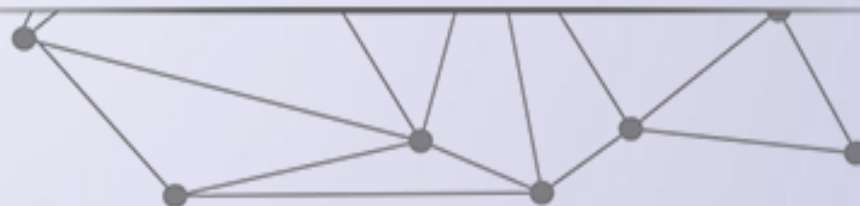




# 面向对象的程序设计



黄天羽

北京理工大学





# 真实世界的对象

- 特征：状态和行为。
- 比如：猫
  - 状态:名字、颜色、品种
  - 行为:喵叫、摇尾巴、捉老鼠





## ■ 真实世界对象

- 这个对象有怎样的状态？
- 这个对象具有什么行为？

## ■ 台灯

- 状态:开、关
- 行为:打开、关闭

## ■ 台式收音机

- 状态：开，关，当前音量，当前频道
- 行为：打开，关闭，增加音量，减少音量，搜索，扫描和调音



# 类：某种类型集合的描述。举例：人

## ■ 属性

- 类本身的一些特性，如名字、身高和体重等属性
- 属性具体值则会根据每个人的不同而不同；

## ■ 方法

- 类所能实现的行为，如吃饭、走路和睡觉等方法。





# 类的定义

```
class classname[(父类名)] : [成员函数及成员变量]
```

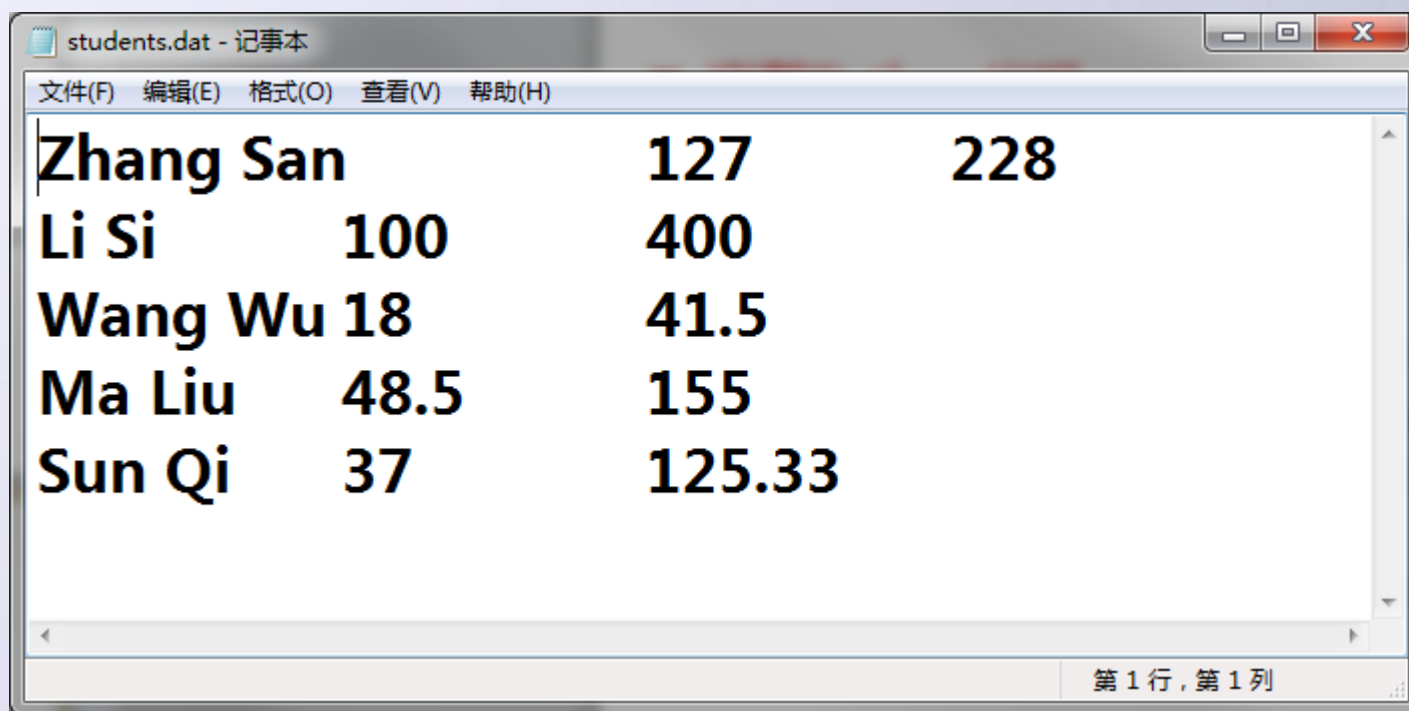
- `_init_`构造函数: 初始化对象的各属性
- `_del_`析构函数: 销毁对象



# 举例：GPA计算

- 学生课程评估：学分和平均绩点GPA。
- 绩点计算以GPA 4分为准则
  - 一门课程3学分
  - 同学得了“A”
  - $3 \times 4 = 12$ 量分数。

记录学生成绩文件students.dat。编写程序，通过读取文件找出平均绩点最高的学生，然后输出他的名字、学分和平均绩点。



The screenshot shows a Notepad window with the title 'students.dat - 记事本'. The menu bar includes '文件(F)', '编辑(E)', '格式(O)', '查看(V)', and '帮助(H)'. The text content is as follows:

Zhang San		127	228
Li Si	100	400	
Wang Wu	18	41.5	
Ma Liu	48.5	155	
Sun Qi	37	125.33	

The status bar at the bottom right indicates '第 1 行, 第 1 列'.



# 定义Student类

```
class Student:
    def __init__(self, name, hours, qpoints):
        self.name = name
        self.hours = float(hours)
        self.qpoints = float(qpoints)
```



# 定义Student类

```
def __init__(self, name, hours, qpoints):  
    self.name = name  
    self.hours = float(hours)  
    self.qpoints = float(qpoints)  
  
def getName(self):  
    return self.name  
  
def getHours(self):  
    return self.hours  
  
def getQPoints(self):  
    return self.qpoints  
  
def gpa(self):  
    return self.qpoints/self.hours
```



# GPA算法描述为

获取文件名

打开文件

设置第一个学生为best

对文件中的每一个学生

if `s.gpa() > best.gpa()`

设置s为best

打印best学生的信息





# 程序运行结果

```
>>>
```

```
Enter name the grade file: students.dat
```

```
The best student is: Sun Qi
```

```
hours: 37.0
```

```
GPA: 3.3872972972972972
```

```
>>>
```

