

## 实验四 PYTHON 字符串与面向对象程序设计

### 一、目的和要求

1. 熟悉 Python 的面向对象定义;
2. 熟悉 Python 的字符串处理;
3. 掌握 Python 语言基本语法;

### 二、实验环境

1. Win 7 操作系统;
2. Python 2.7.X, IDLE、PyCharm 等开发环境;

### 三、实验内容

#### (一) 验证实验 (每个同学完成)

1. 运行调试第四章各小节例示代码及课后练习的程序设计题, 检查运行结果是否正确, 记录实验结果。
2. 运行调试第六章各小节例示代码及课后练习的程序设计题, 检查运行结果是否正确, 记录实验结果。
3. 阅读和运行 `reducible.py`, 比较该代码和你的小组在实验三中所完成的查找可缩减单词代码查找的效率。
4. 字符串的验证: 编制系列单词处理函数, 分别实现下述功能, 并验证正确性。
  - (1) 编写函数 `rotateword`, 接收一个字符串 `strsrc` 以及一个整数 `n` 作为参数, 返回新字符串 `strdes`, 其各个字母是 `strsrc` 中对应位置各个字母在字母表中“轮转” `n` 字符后得到的编码。
  - (2) 编写函数 `avoids`, 接收一个单词和一个含有禁止字母的字符串, 判断该单词是否含有禁止字母。
  - (3) 编写函数 `useonly`, 接收一个单词和一个含有允许字母的字符串, 判断该单词是否仅仅由允许字母组成。

(4) 编写函数 `useall`，接收一个单词和一个含有需要字母的字符串，判断该单词是否包含了所有需要字母至少一个，并输出 `words.txt` 中使用了所有元音字母 `aeiou` 的单词。

(5) 编写函数 `hasnoe`，判断一个英语单词是否包含字母 `e`，并计算 `words.txt` 中不含 `e` 的单词在整个字母表中的百分比。

(6) 编写函数 `isabecedarian`，判断一个英语单词中的字母是否符合字母表序，并且输出 `words.txt` 中所有这样的单词。

## 5. 面向对象编程的验证

(1) 阅读和运行 `Kangaroo.py`，调用和测试其种所定义的类 `Kangaroo` 的方法，分析方法实现中的 `bug`，修正，写入实验报告。

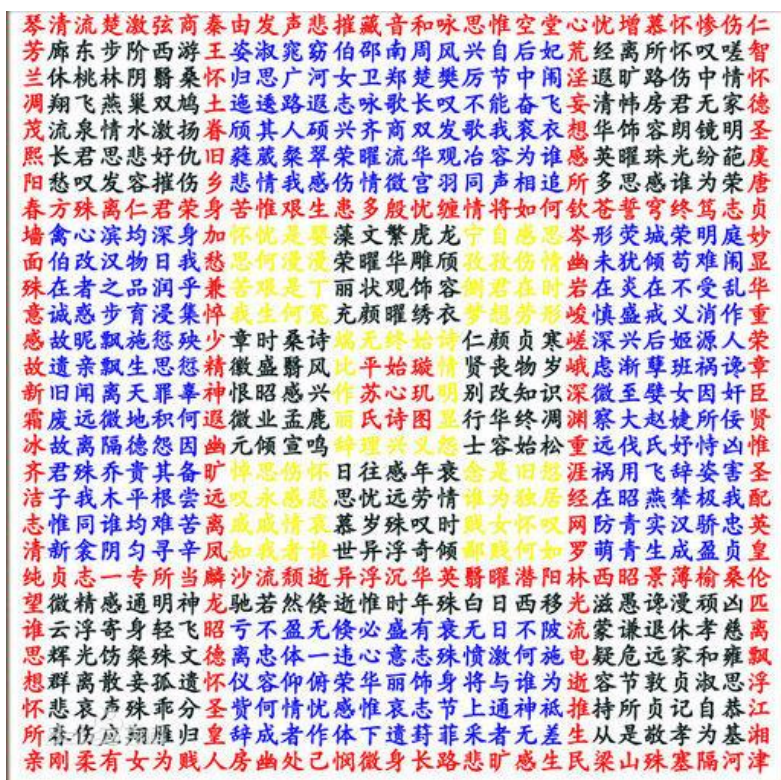
(2) 阅读和运行实现了扑克牌游戏部分功能的 `Card.py` 和 `Pokerhand.py`，理解 `Card`，`Deck`，`Hand` 和 `Pokerhand` 四个类的功能差别和类之间的调用关系，写入实验报告。

## (二) 设计实验（小组完成，代码提交，算法设计和测试写入实验报告）

6. 编写拥有 a、对象成员 `hour`，`minute` 和 `second` 的时间类 `Time`；b、重载 `__str__` 和 `__add__` 方法；c、方法 `time2int`：把时间对象转换为秒数；d、方法 `printtime`：输出时间；e、方法 `isafter`：判断两个时间对象的先后；f、方法 `increment`：计算对象经过 `n` 秒后时间；g、方法 `isvalid`：判断时间对象合法性。在主函数设计代码验证 `Time` 各个方法的正确性。

7. “璇玑图”总计八百四十一字，纵横各二十九字，纵、横、斜、交互、正、反读或退一字、迭一字读均可成诗，诗有三、四、五、六、七言不等，目前统计可以组成 7985 首诗。运用 Python 字符串处理的方法，解读“璇玑图”，输出尽可能多的诗词，文本原文和解读方法可参照下面链接或者其他课外资料：

<http://www.uus8.org/4/%CB%C4%BF%E2%C8%AB%CA%E9%C0%E0/%BC%AF%B2%BF/%C6%E4%CB%FB/%E8%AF%E7%E1%CD%BC%CA%AB%B6%C1%B7%A8.htm>）。



### (三) 综合实验（小组完成，代码提交，算法设计和测试写入实验报告）

#### 7. 马尔可夫文本分析和应用

(1) 马尔可夫文本分析计算文本中单词组合和其后续单词（含标点符号）的映射，这个单词组合被称为马尔可夫分析的前缀，前缀中单词的个数被称为马尔可夫分析的“阶数”。编写 Python 代码实现某个文本的  $n$  阶马尔可夫文本分析，并且将分析结果记录在字典中。

(2) 采用所实现的马尔可夫分析，对“emma.txt”或“whitefang.txt”进行马尔可夫分析，运用  $n$  阶马尔可夫分析的结果生成由  $m$  个句子（注意首字母大写和结尾标点符号）组成的随机文本。分析所生成文本的语义自然性和阶数  $n$  的关系。

(3) 尝试采用 Python 不同的序列数据结构表示前缀，比较运行效率的差异。

#### 8. 模拟快餐订餐场景

(1) 定义 4 个类：Customer 顾客类，Employee 商户类，Food 食物类 以及 Lunch 订餐管理。

(2) Lunch 类包含 Customer 和 Employee 实例，具有下单 order 方法，该方

法要求 Customer 实例调用自身的 placeOrder 向 Employee 对象要求下单, 并且获得 Employee 对象调用 takeOrder 生成和返回一个 Food 对象, Food 对象应当包含了食物名字字符串。调用关系如下:

Lunch.order—› Customer.placeOrder—› Employee.takeOrder—› Food

(3) Lunch 类包含 result 方法, 要求 Customer 打印所收到的食物订单。

(4) 编写交互式界面验证所设计的订餐系统。