RAT程序说明文档

1. UML类图
2. 设计思路

使用Dictionary，Word，Entry三个层次作为词典的基础。其中entry继承出很多不同了实现类，因为不同词性的义项包含的内容是不同的。其中Dictionary使用单件模式，仅在程序开始时自动构造一次，并调用read方法来读取资源文件中的内容，之后不能再次被构造。而Word，Entry的创建使用工厂模式，因为创建的过程十分复杂而且要判断Entry具体的实现类类型。

软件支持多个用户，每个用户拥有自己的set，这是一些属于Dictionary的单词的集合，用户的test和recite功能都是针对set而不是全部的单词（也可以把全部单词作为一个set）。Class User表示用户。Class set 表示集合。User中有vector<set\*>的属性保存用户的全部set。

Class interface表示界面。实现类consoleInterface表示命令行界面。用户的命令将在某些方法中进行分析然后传递给consoleInterface的其他方法进行执行。这个类承担所有和用户交互的工作。

Class test :

Class recite:

1. 设计说明
2. Dictionary

属性包括Dictionary的一个实例（static）以及vector<Word>。

方法包括进行精确查询，匹配查询，子串匹配（kmp算法），插入，排序，读取文件等。

1. Word

属性有单词的拼写，vector<entry\*>,背诵的有关参数。

提供的方法包括排序，查找，判断一个单词是否熟悉等。重载了向屏幕输出的<<。

1. Entry

根据不同的词性有不同的属性。比如名词会记录是否是复数，是否是缩写，复数形式等。

重载了向屏幕输出的<<运算符，提供向资源文件输出的方法。

1. Factory

包括WordFactory的EntryFactory（抽象类）。主要承担分析资源文件并且生成product的过程。

1. Interface

属性包括用户，当前用户，以及一些运行参数（比如到目前是否有数据被修改）。

提供众多的方法：初始化，接受用户命令，分析用户命令，查询，测试，背诵......大部分通过对命令的分析调用其他类中的方法。用户登录，切换用户等是直接在这里实现了。

1. User

属性包括每个用户的用户名，密码，等级，vector<set\*>等。构造函数将从文件中读取用户的数据。

1. Set

包含vector<Word\*>。

方法包括查找，插入等。

1. Test
2. Recite