**深 圳 大 学 实 验 报 告**

**课程名称： 面向对象技术与方法**

**实验项目名称： 实验4数据预处理与类信息统计**

**学院： 计算机技术与软件学院**

**专业： 计算机技术**

**指导教师： 尹剑飞**

**报告人： 汪乔 学号： 2172272037 班级：**

1. 设计说明

（至少覆盖：UML类图+说明、时序图（每个功能一张时序图）+说明、数据结构设计说明）。设计说明需回答做出关键的设计决定的理由或依据是什么，为什么要这样设计，并且回答以下问题：



其中，在Record类中含有四个规范化操作的方法未写在上面。

下图是在实验一的基础上对record进行规范化后保存到文件中。



下图是实验二部分。



数据结构设计说明：

Folder类：

属性：

file\_path:：文件夹名

files：文件夹下所有的文件对象

方法：

save\_all()：用于对实验一的数据进行操作

file\_new()：对实验二的数据进行操作

\_write\_first\_row()：在实验二中写入第一行，同时有初始化文件的功能；

\_has\_next()：判断文件是否全部读完；

\_get\_same\_record()：对所有的文件进行读取，直到所有的记录的日期相同；

StrMatrix类：

属性：

row：矩阵的行数

col：矩阵的列数

row\_set：矩阵的数据

方法：

\_prep：进行一些初始化操作，可以放到\_\_int\_\_中；

fill\_one\_point：填充矩阵的一个值

fill\_one\_col：填充矩阵的一列值

fill\_cols：填充矩阵的多列值

write\_lines：将字符串矩阵自身写入到文件中

file类：

\_file\_point：文件指针

\_label\_num：标签号

file\_name：文件名

方法：

add\_record()：将有效记录添加到列表里

get\_attribute()：得到数据集的属性

has\_next()：判断该文件是否读取完成

save()：实验一的主要逻辑代码，得到均值等数据，再一行行处理record

\_get\_max\_precison()：得到最大精度

\_sys\_to\_num()：将字符串转换成数字

Record类：

属性：

\_label：标签

record\_data：除标签外的数字

\_fie\_name：文件名

方法：

read\_line()：读取一行，得到label和record\_data

add()：进行错误检查，如果正确，添加到列表中，返回True，否则返回False

get\_label()：如果是日期则格式化，返回标签

save()：在实验一中，将自身保存到文件中

normalization()：规范化方法的控制函数，调用下面四个规范化方法

\_z\_score\_scaling()、\_dividing\_standard\_deviation()、

\_0\_1\_scaling()、\_dividing\_each\_value\_by\_the\_range()

\_has\_exception()：检查异常，如果有则返回True

1. 为什么要定义这样一个类，这个类名命名是否体现单一责任原则、名词化、拟人化、功能完整性、权力下放、对象自治性，怎么体现？

Folder/File/Record类是根据数据来定义的类，StrMatrix类是保存一个日期的所有记录。方法实现是就近原则，主要写在最靠近的类下，像读取一行、添加到列表中都是在Record类中完成。

1. 每个类的公有方法是否控制在很少的范围（如公有方法不超过10个），公有方法越少，则越方便用户使用，同时越能体现类的单一责任原则。每私有方法是否都相应地支撑公有方法。

由于功能简单，公有方法控制在很少的范围内。

1. 每个方法体的复杂性是否何控制的，包括每个方法体的代码行数是否控制在200行以内，循环深度是否控制在2层以内？

主要是通过将逻辑拆分，放在下层类中去实现，使用了python的一些语言特性，用于减少代码行数。存在少量循环深度大于2层，大部分是分开写的。

1. 对象之间的交互（消息传递）是否过于复杂，对象之间的交互包括对象之间是如何建立引用关联的（指针关联的）？，传递的参数是否过多？，传递的参数是否包括不必要的、每次都需要传递的参数？

为了降低交互的复杂度，简化了交互流程，在读取记录上可以体现：当数据存在异常时，继续往下读取，直到得到正常的，之前存在在Folder类里还需要判断的情况；将record\_data属性去除了label，方便方法使用。少部分存在传参的个数较多的情况。

1. 面向对象编程是有状态编程，每个对象是否都保留有它应该管理的数据成员（状态），它管理的数据成员不多（即具有单一责任属性）、也不少（即保证对象的数据完整性）。
2. 每个类的方法是否都尽可能只依赖该类的成员变量？，还是需要结合内部成员变量与外部传入的参数？为每个类统计一下：该类有多少方法需要结合传递notrivial参数才能正常工作？当这种类型的方法占该类总方法数 > 20%时，这个类的方法是否应该合并到调用方所在的类中成为私有方法，为什么？
3. 为你所写的每个类，统计一下：每个类的代码行数、共有多少方法、共有多少属性、共有多少外部传入的参数用于完成这个类的功能、每个方法有多少行代码。作出一张列表：|类名 ｜ 方法数量 ｜ 属性数量 ｜ 最大方法体的代码行数 ｜ 最小方法体的代码行数 ｜ 类中所有方法代码行数分布情况（平均值、中位数、标准差） ｜ 外部传入的参数数量 ｜ ，并分析这张表中，哪些类的责任过载？哪些类的责任过轻？如何分析？、如何改进？
4. 统计类与类之间的消息交互量，做类似协方差矩阵：

类1 ｜ 类2 ｜ … | 类n

类1 类1私有函数调用次数

类2 类1与类2消息调用次数

.

.

.

类n

利用上表，分析类之间的协作密集性，从而决定是否需要合并类，还是拆分类。

1. 可能的改进说明

存在除零现象，可以在record方法中，将四个规范化方法改进。

1. 实验总结

（此处可以陈述自己在完成该实验时遇到的困难和弯路，以及可能的进步方向等。）