**中山大学计算机学院**

**人工智能**

**本科生实验报告**

**（2024学年春季学期）**

课程名称：Artificial Intelligence

|  |  |  |  |
| --- | --- | --- | --- |
| 教学班级 | **202320346** | 专业（方向） | **计算机科学与技术** |
| 学号 | **22320107** | 姓名 | **饶鉴晟** |

# 实验题目

给定student\_data.txt文本文件，每一行是一名学生的信息，从左到右分别是该学生的姓名，学号，性别和年龄，每个属性以空格间隔。数据类型如下：

name：str # 姓名

stu\_num：str # 学号

gender：str # 性别，”M”为男性，”F”为女性

age：int # 年龄

编写StuData类，须有以下方法：

1. 构造函数（\_\_init\_\_），以文件名（str类型，带.txt后缀）为输入，读取文件中的学生信息，储存到类成员data中。data的数据类型为list，其中每一个学生的信息以列表方式存储。

例如，读入一行学生信息”Aaron 243 M 18”，则data变为

[[“Aaron”, “243”, “M”, 18], [“Eric”, “249”, “M”, 19]]

2. AddData方法，以单个学生的信息作为输入，储存到data中。调用该方法的参数形式为学生属性的4个关键字实参。例如，执行self.AddData(name=”Bob”, stu\_num = “003”, gender=”M”, age=20)后，data变为

[[“Aaron”, “243”, “M”, 18], [“Eric”, “249”, “M”, 19], [“Bob”, “003”, “M”, 20]]

3. SortData方法，以学生某个属性（str类型，是'name','stu\_num','gender','age'的其中之一）作为输入，将data按该属性从小到大排序。可以假定不会输入非学生属性的字符串。例如，执行self.Sort('stu\_num')后，data的学生信息按学号从小到大排序，变为

[[“Bob”, “003”, “M”, 20], [“Aaron”, “243”, “M”, 18], [“Eric”, “249”, “M”, 19]]

4. ExportFile方法，以导出的文件名（str类型，带.txt后缀）为输入，新建一个txt文件，将data中的数据按当前列表顺序导出到该文件内，格式同原student\_data.txt文本文件，即"姓名学号性别年龄"，并存储到目标文件夹。例如，调用self.ExportFile('new\_stu\_data.txt')，则将data中数据导出到new\_stu\_data.txt文件。

# 实验内容

1. 算法原理

这段代码实现了一个学生信息管理系统的基本功能，包括从文件中读取学生信息、添加新的学生信息、根据指定属性对学生信息进行排序以及将学生信息导出到文件中。

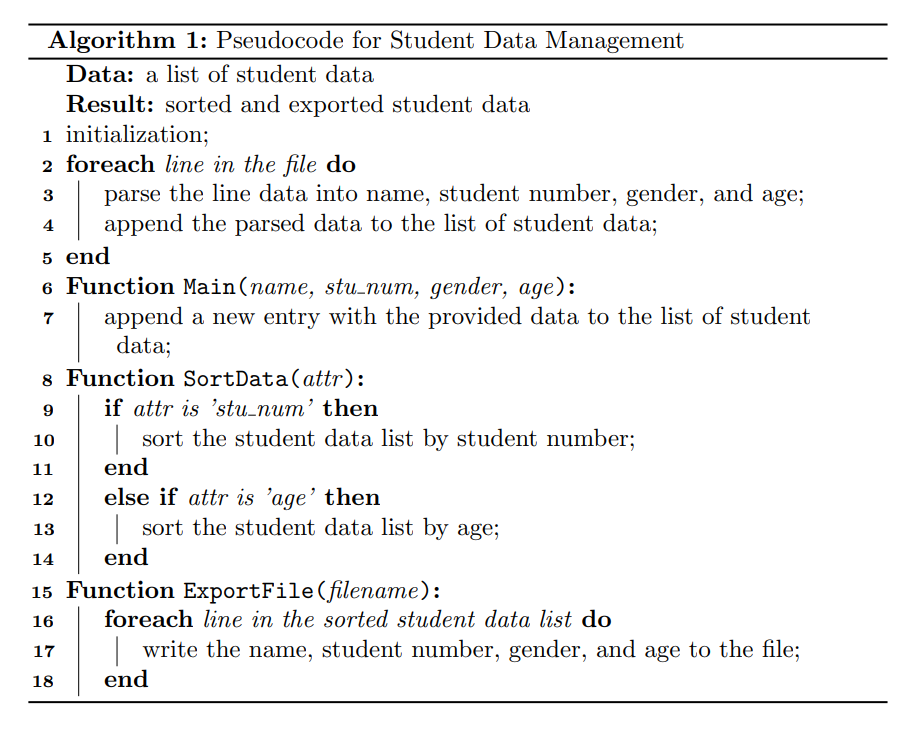
1. 读取文件并解析数据：首先，程序会打开指定的文件，逐行读取文件内容。对于每一行数据，使用空格分割，然后将姓名、学号、性别和年龄分别存储到临时列表 line\_data 中，并将其添加到 self.data 中，形成一个二维列表，存储了所有学生的信息。

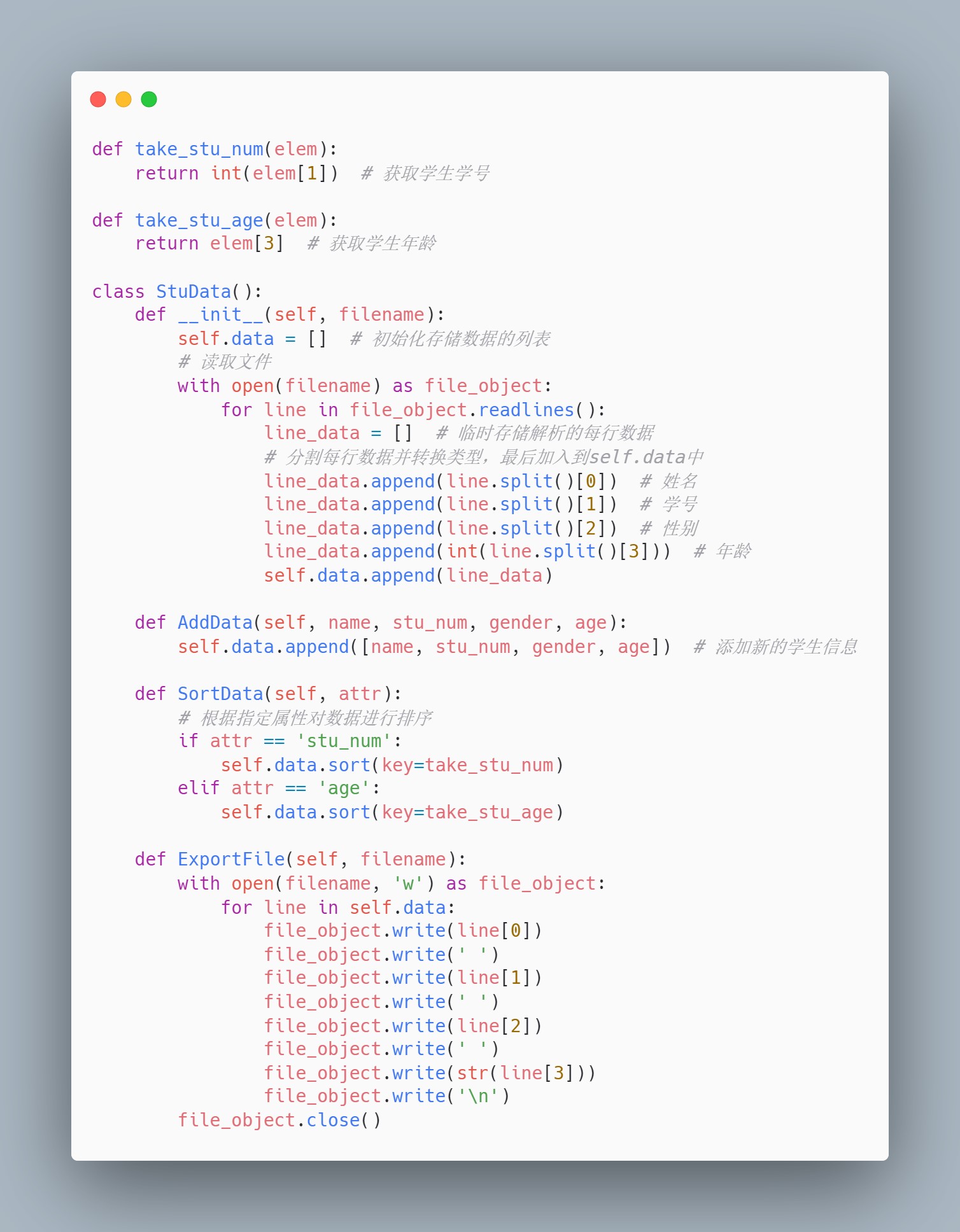
2. 添加新的学生信息：通过 AddData 方法，可以向学生信息列表 self.data 中添加新的学生信息。这个方法接受姓名、学号、性别和年龄作为参数，将这些信息组成一个列表并添加到 self.data 中。

3. 根据指定属性排序：SortData 方法允许根据学号或年龄对学生信息进行排序。如果指定排序属性为学号 (stu\_num)，则调用 take\_stu\_num 函数进行排序；如果指定为年龄 (age)，则调用 take\_stu\_age 函数进行排序。排序完成后，学生信息列表 self.data 会按照指定属性的顺序重新排列。

4. 导出学生信息到文件：通过 ExportFile 方法，可以将学生信息导出到指定的文件中。该方法会打开指定文件，遍历学生信息列表 self.data，将每个学生的姓名、学号、性别和年龄以指定格式写入文件中，每个学生信息占据一行，以换行符分隔。

整体算法的实现思路主要包括文件读取与解析、添加数据、排序以及文件导出。通过这些功能，可以方便地管理学生信息，并对学生信息进行排序和导出操作。

1. 伪代码
2. 关键代码展示（带注释）



# 实验结果及分析

1. 实验结果展示示例（可图可表可文字，尽量可视化）

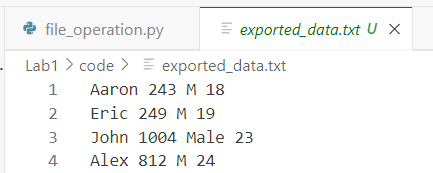


**测试程序输入样例**

散点图

描述已自动生成

**测试程序输出样例**

****

**测试程序输出文件内容样例**

**如图所示，程序正确初始化数据，可以以指定字段对学生数据进行排序，并且可以正确输出数据到指定文件当中。**

# 思考题

**1. 如果用列表作为字典的键，会发生什么现象？用元组呢？**

答：

列表不可以作为字典的键，如果强行将列表作为字典的键，python解释器会报错"TypeError: unhashable type: 'list'"，原因是因为python的字典类型要求其键必须是不可变类型，类似C/C++语言当中的右值，而在python当中列表是一种可变类型，相当于C/C++当中的左值，可变类型在更改其内容的时候其哈希值会变，导致python无法将定义时的键与值链接，所以会报错。

如果用元组的话不会报错，程序可以正常运行，因为元组是一种不可变类型，拥有固定的哈希值，因此可以作为python的键。

**2. 在本课件第2章和第4章提到的数据类型中，哪些是可变数据类型，哪些是不可变数据类型？试结合代码分析。**

答：

不可变类型：数字，字符串，空值，元组（均可以作为右值）

可变类型：列表，字典，集合（均可以作为左值）