

Lab1实验要求

使用简化的 **Markdown** 语法来创建和编辑文件。语法包括使用 **#**, **##**, **###** 等作为不同层级的标题项，以及使用 *****, **-**, **+** 等无序列表或 **1.**, **2.** 等有序列表作为文本项。

基本功能

1 文件操作

命令 1-1：**load** 文件路径

功能描述：

从指定路径加载文件到内存中进行编辑。如果指定的文件不存在，则新建一个文件。文件路径应该是合法的，不能含有非法字符。在已经加载文件后继续载入新的文件，则将原先的文件保持在后台，自动切换到新加载的文件进行编辑。不能重复打开同一个文件。

输入示例：

```
load test.md
```

输出示例：无

命令 1-2：**save**

功能描述：

将内存中的当前编辑数据保存到之前加载的或新创建的文件中。确保有一个文件已经被加载或创建。

输入示例：

```
save
```

输出示例：无

命令 1-3：**ws**

功能描述：

显示已加载的所有文件名称，按加载顺序排列并显示序号。修改未保存的文件后加上*符号（包括新建的文件，和加载后已经修改过的文件），当前正在编辑的文件加上<符号。

输入示例：

```
load newfile.md
WS
```

输出示例：

```
1 test.md
2 newfile.md*<
```

命令 1-4：**switch** 文件序号

功能描述：

切换到指定序号的文件进行编辑，对应序号必须存在。

输入示例：

```
switch 1
WS
```

输出示例：

```
1 test.md<
2 newfile.md*
```

命令 1-5：**close** 文件序号

功能描述：

关闭指定序号的文件，对应序号必须存在。未保存的文件会询问是否保存：**save changed file before closing? (y/n)**，若输入y则执行保存；若输入n直接关闭，不会写入磁盘。关闭当前文件时，当前正在编辑的文件变为上一个序号的文件，若没有则变为下一个序号，若为唯一文件则返回到未加载文件的状态。

输入示例：

```
close 2
n
```

```
WS
```

输出示例：

```
save changed file before closing? (y/n)
```

```
1 test.md<
```

2编辑功能

命令2-1：**insert** [行号] 标题/文本

功能描述：

行号参数可选。

在指定行插入标题或文本。如果不指定行号，则默认在文件的最后一行插入内容。

输入示例：

```
insert 3 ### 新的子标题  
list
```


输出示例：

```
# 我的书签  
## 学习资源  
### 新的子标题  
### 编程  
* Java从入门到入土
```

命令2-2：**append-head** 标题/文本

功能描述:

在文件起始位置插入标题或文本。

 Tips: 相当于insert 命令的特殊变体

输入示例:

```
append-head # 新的标题
list
```


输出示例:

```
# 新的标题
# 我的书签
## 学习资源
### 新的子标题
### 编程
* Java从入门到入土
```

命令2-3：**append-tail** 标题/文本

功能描述：

在文件最后一行插入标题或文本。

 Tips: 相当于insert 命令的特殊变体

输入示例:

```
append-tail * 新的文本
list
```

输出示例：

```
# 新的标题
# 我的书签
## 学习资源
### 新的子标题
### 编程
* Java从入门到入土
* 新的文本
```

命令 2-4：**delete** 标题/文本 或**delete** 行号

功能描述：

删除指定标题或文本。如果指定行号，则删除指定行。当删除的是标题时，其子标题和内容不会被删除。

输入示例：

```
delete Java从入门到入土
```

输出示例：

```
# 新的标题
# 我的书签
## 学习资源
### 新的子标题
### 编程
* 新的文本
```

命令 2-5：**undo**

功能描述：

撤销上一次执行的编辑命令，返回到执行该命令前的状态。不适用于非编辑命令。上一个命令属于**load**与**save**等文件相关命令组时不能被撤销不能被跳过，属于**list**，**list-tree**与**dir-tree**等显示相关命令组时应该被跳过。

输入示例：

```
undo
list
```

输出示例：

```
# 新的标题
# 我的书签
## 学习资源
### 新的子标题
### 编程
* Java从入门到入土
* 新的文本
```

命令 2-6：**redo**

功能描述：

重新执行上次撤销的命令，内容返回到执行该命令后的状态。即只有上一个编辑命令是**undo**时，才允许执行**redo**。

输入示例：

```
redo  
list
```

输出示例：

```
# 新的标题  
# 我的书签  
## 学习资源  
### 新的子标题  
### 编程  
* 新的文本
```

3显示功能

命令 3-1：`list`

功能描述：

以文本形式显示当前编辑的内容。

输入示例：

```
list
```

输出示例：

```
# 新的标题  
# 我的书签  
## 学习资源  
### 新的子标题  
### 编程  
* 新的文本
```

命令 3-2：`list-tree`

功能描述：

以树形结构显示当前编辑的内容。

输入示例：

```
list-tree
```

输出示例：

```
├─ 新的标题
├─ 我的书签
│   └─ 学习资源
│       ├── 新的子标题
│       └─ 编程
│           └─ .新的文本
```

命令 3-3：**dir-tree** [目录]

功能描述：

目录参数可选。

以树形结构显示指定目录（标题）下的内容。如果不指定目录，默认显示当前工作目录下的内容，即完整的树结构。

输入示例：

```
dir-tree 学习资源
```

输出示例：

```
└─ 学习资源
    ├── 新的子标题
    └─ 编程
        └─ .新的文本
```

日志模块

记录每一次命令，包括执行的时间戳。同时将每次运行中（Session）的命令历史分段保存在一个日志文件中，退出交互界面后依然可以通过日志查看命令历史。如果 history 执行失败仅仅提示警告，不影响其他功能。

日志文件格式：

```
session start at YYYYMMSS HH:mm:ss
//...
session start at 20230926 14:52:36
```

```
20230926 14:53:25 load books.md
20230926 14:55:43 dir-tree 学习资源
session start at 20230928 18:22:56
20230928 18:23:22 load traval.md
20230928 18:23:35 append-head 旅行清单
20230928 18:25:40 save
```

命令：**history** [数量]

功能描述：

数量参数可选。

列出指定数量的最近历史命令，包括执行命令的时间戳。默认显示全部记录，但可以通过参数限制显示的数量。如果指定的数量大于存储的历史命令数量，也显示全部命令记录。

输入示例：

```
history 4
```

输出示例：

```
20230928 18:25:40 save
20230928 18:23:35 append-head 旅行清单
20230928 18:23:22 load traval.md
20230926 14:55:43 dir-tree 学习资源
```

统计模块

实时在一个日志文件中记录一次运行中 (Session) 打开的md文件以及在各个文件上工作的时长。

1. 可以使用**stats**得到当前运行过程中的统计信息
2. 通知记录到日志中，日志保存了历次的操作
3. 当前编辑文件在没有关闭之前，不必将当前文件的统计信息实时计入日志，当前文件的统计只能通过**stats current**得到。
4. 如果执行 **stats** 命令失败仅仅提示警告，不影响其他功能继续执行。

日志文件格式：

```
session start YYYYMMSS HH:mm:ss
//...
session start YYYYMMSS HH:mm:ss
./test.md 12 分钟
```



```
./notes.md 2 小时 15 分钟  
./draft.md 45 分钟
```

命令：`stats [all | current]`

功能描述：

all 或 current 参数可选。

显示当前会话中指定文件或所有文件的工作时长统计。默认为 `current`，显示当前文件的统计信息。根据不同时长，以可读的方式显示不同的时长单位。例如，65 分钟显示为 1 小时 5 分钟。

输入示例：

```
stats all
```

输出示例：显示当前会话中所有**Markdown**文件的工作时长

```
session start YYYYMMSS:HHmmSS  
./test.md 12 分钟  
./notes.md 2 小时 15 分钟  
./draft.md 45 分钟
```

时长单位参考：

1. 秒 (Second) 时间段：1秒
2. 分钟 (Minute) 时间段：60秒 = 1分钟
3. 一刻钟 (Quarter of an hour) 时间段：15分钟 = 一刻钟
4. 半小时 (Half an hour) 时间段：30分钟 = 半小时
5. 小时 (Hour) 时间段：60分钟 = 1小时
6. 天 (Day) 时间段：24小时 = 1天

注意事项

1. 两种树形结构的显示应尽可能复用代码。
2. 日志模块和统计模块是相对不稳定的，可能会迭代的功能，核心的模块尽可能避免与这两个模块直接耦合。
3. 编辑前需要先加载文件，同一个会话内可以加载并编辑不同的文件：文件操作 → 编辑 | 统计 → 文件操作 → 编辑 | 统计 →
4. 将无序列表和有序列表视为文本项，统一看作叶子节点，暂时不用考虑输入多级列表。

5. 假设所有输入的标题都是从一级标题逐级递增的，不会出现跨层级添加标题的情况，暂时不考虑测试语法的合法性(即假设不会产生非法的构造)。
6. 虽然不用考虑会产生非法的树结构，但对于一些异常输入要有异常处理，不能因为越界、空指针等问题导致程序直接崩溃，即不能被动中断会话。

提交要求

提交文件说明

- **源代码**：代码应当尽量整洁，包含（自动化）测试用例。自动测试用例可借助JUnit工具(如JUnit等)来编写。
- **功能完成度列表**：列出所有要求功能，并标注其完成状态（完成/未完成/部分完成），可附上截图。
- 一份简单的**使用说明**：只需要包括运行环境、启动步骤，并说明使用了哪些设计模式。
- 所有提交的文件都应该被打包在一个名为 **lab1-学号-姓名** 的 **zip** 压缩文件中。

提交时间

- 截止时间：2023 年 12 月 1 日 23 点 59 分 59 秒。

其他注意事项

- 此实验为**个人作业**。
- 可采用任意编程语言。
- 灵活运用设计模式以复用代码，降低模块间的耦合度。
- 在截止时间之前**可以多次提交**。