

# Lab2 实验要求

---

本实验需要在Lab1的项目中扩展以下功能

## 扩展功能

---

### 1 使用自动化测试验证设计和实现

---

使用xUnit工具构建已完成的部分的单元测试代码

- 根据讨论时分层的建议，对各层单独测试。
- 写日志时写入了系统时间。需要在统计模块的单元测试中验证时否写入了时间。

### 2 支持多个workspace

---

每次装入文件后会生成一个新的workspace。但同时只有一个活动的workspace，每个workspace都可以独立进行Lab1规定的操作。（参考IDE中同时打开多个文件的行为）。

#### 命令2-1: `load` 文件路径

功能描述：

1. 从指定路径加载文件到内存并自动创建workspace，并将该workspace设为活动状态，可进行编辑
2. 如果指定的文件不存在，则新建一个文件。
3. 如果该文件已经在某个workspace中打开，则不能重复加载

输入示例：

```
load test1.md
```

输出示例：

无

#### 命令2-2: `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2-3: `list-workspace`

### 功能描述:

展示当前所有已经打开过的workspace，在活动的workspace前加 `->` 标记。若有workspace修改过但未保存，展示该workspace时需在名字后加\*。

### 输入示例:

```
load test1.md
append-head # 新的标题1
save
load test2.md
append-head # 新的标题2
list-workspace
```

### 输出示例:

```
test1
->test2 *
```

## 命令2-4: `open`

切换活动的workspace

1. 多个workspace中只有当前的workspace处于active状态。
2. 从当前workspace切换到另一个workspace时即使当前workspace的内容未保存，再切换回来时也可以继续当前的操作。但在exit后未保存的workspace的操作则丢失。

如果在 `list-workspace` 时输出:

```
test1
->test2 *
```

接着输入命令 `open test1`，  
再次 `list-workspace` 的输出为

```
->test1
test2 *
```

接下来的所有操作在 `test1`上执行

## 命令2-5: `close`

### 功能描述:

关闭当前正在编辑的文件，若未保存则提示是否保存，选择“Y”后保存并关闭此工作区，选择“N”后丢弃本次编辑的内容并关闭此工作区。工作区关闭后主程序没有退出，还可以继续使用命令打开新的工作区。但此时再执行`list-workspace`则不会出现该workspace。

输入示例:

```
close
```

输出示例:

```
Do you want to save the current workspace [Y\N] ?
```

## 命令 2-5: `exit`

功能描述:

直接退出整个程序，退出时若有workspace的内容未保存，则需询问是否要保存所有workspace的内容？若不保存则丢失该workspace之前操作的内容。

输入示例:

```
load test1.md
append-head # 新的标题1
save
load test2.md
append-head # 新的标题2
exit
```

输出示例:

```
Do you want to save the unsaved workspace [Y\N] ?
```

## 3 显示当前工作目录的内容

以树形结构展示现在load文件的默认相对目录，这个目录下可以有子目录。

### 命令 3-1 `ls`

输入示例:

假设当前load文件的默认相对目录中有test1.md与data文件夹两个内容。

```
ls
```

输出示例:

```
|— test1.md  *
|— data
   |— data1.md
```

其中，末尾带\*表示该文件已经在某个工作区中打开。

## 4 工作区恢复

**注意：**需要用到我们没有讲过的设计模式Memento，需要大家自行学习。

假设上次退出时保存了所有的工作区文件，并且再次启动前工作区内的文件不会被修改。再次启动程序时，自动恢复上次退出时所有工作区的状态，包括：

1. 打开上次退出时所有的工作区
2. 将上次退出时所在的活动工作区，设为当前的活动工作区
3. (选做，加分项：5%) 对于某个恢复的工作区(某个文件)，同时恢复所有工作区的redo/undo状态，也就是说，在退出前这个工作区undo/redo是什么行为，恢复后是同样的行为。

## 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。
- 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。
- 所有提交的文件都应该被打包在一个名为 lab2-学号-姓名 的 zip 压缩文件中。

### 提交时间

- 截止时间：2023 年 12 月 3 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。