

# Lab Bookmark Editer

## 总体说明

本次Lab的目标：

1. 恰当地建立面向对象的模型。
2. 使用恰当地设计模式增加程序的灵活性。
3. 使用自动化测试验证设计和实现。

## 内容

### 基于命令行的网站书签编辑器

实现一个简化的、网站书签管理器，书签栏的结果保存在一个Markdown格式的文件中。该Markdown中只有两种markdown语法。

#### 1. 分级标题

```
# 标题1
## 标题2
### 标题3
```

#### 2. 链接

```
[Markdown Guide](https://www.markdownguide.org)
```

一个书签栏的数据样例如下：

```
# 个人收藏
## 课程
[elearning](https://elearning.fudan.edu.cn/courses)
## 参考资料
[Markdown Guide](https://www.markdownguide.org)
### 函数式
[JFP](https://www.cambridge.org/core/journals/journal-of-functional-programming)
### 面向对象
## 待阅读
[Category Theory](http://www.appliedcategorytheory.org/what-is-applied-category-theory/)
```

注意：

Markdown只是书签栏的一种简化的存储(Persistence)方式，以后缀为bmk的文本文件的形式存储。

## 1.2 显示书签栏的树形结构

通过字符界面以树形结构的方式显示当前工作空间的书签栏的编辑结果

## 1.3 显示当前的文件目录结构

通过字符界面显示文件系统的目录结构

# 功能说明

## add 命令

程序启动后，在命令行可以以如下的方式建立样例中的书签栏：

```
add-title "个人收藏"
add-title "课程" at "个人收藏"
add-title "参考资料" at "个人收藏"
add-title "函数式" at "参考资料"
add-title "面向对象" at "参考资料"
add-title "待阅读" at "个人收藏"
add-bookmark "elearning"@https://elearning.fudan.edu.cn/courses at "课程"
add-bookmark "Markdown Guide"@https://www.markdownguide.org at "参考资料"
add-bookmark "JFP"@https://www.cambridge.org/core/journals/journal-of-functional-programming at "函数式"
add-bookmark "Category Theory"@http://www.appliedcategorytheory.org/what-is-applied-category-theory/ at "待阅读"
```

注意：如果运行 add 命令时 at 的标题在文件中重复出现，则该 add 命令生效的标题为第一个出现的标题。

## delete 命令

可以删除某项内容

```
delete-title "参考资料"
delete-bookmark "Markdown Guide"
```

注意：如果有重复的标题需要把所有匹配的内容都删除。对于 delete-title，如果是父节点，其子节点也一起删除。

## 装入命令

可以从文件中装入书签栏

```
open "./path/to/cloud.bmk"
```

路径的写法可以根据操作系统自行选择格式。

此时将工作空间的内容替换为指定文件中的内容，并可以继续编辑。如果文件不存在，认为是新的空白文件。

启动程序时需要在命令行参数中指定一个文件名，比如

```
bookmark "./path/to/cloud.bmk"
```

这相当于启动程序后直接运行了命令

```
open "./path/to/cloud.bmk"
```

每次打开/新建一个文件，可以认为开辟了一个新的工作空间，并且将原来的工作空间完全废弃掉。

## 保存命令

可以将结果保存

```
save
```

根据具体情况，将覆盖原有的文件或者生成新的文件。

## undo / redo

对于add-\* 和 delete-\* 指令，编辑的过程中支持undo和redo。

```
undo  
redo
```

## 可视化地显示当前编辑的内容

使用show-tree命令随时可以显示当前工作空间中书签编辑的结果

```
show-tree
```

## 可视化地显示当前目录结构

使用ls-tree命令可以随时显示当前目录下的文件的树形结构

```
ls-tree
```

**注意：**

恰当地选择和使用设计模式(适配器模式)，尽可能共享上面的两种显示tree的代码。树形结构的显示方式参考如下。

- 显示格式：使用双引号来区分书签和标题（或文件名和目录），也即在使用 show-tree 时下例中 B1 为书签名，其余为标题名；而在使用 ls-tree 时 B1 为文件名，其余为目录名。
- 显示顺序：show-tree 命令下同级的打印顺序与标题或者书签的添加先后顺序相同；ls-tree 命令下同级的打印顺序不作要求。

```
└─ A  
  └─ B  
    │ └─ "B1"  
    │ └─ B2  
  └─ C  
  └─ D  
    └─ D1
```

# 访问指定的书签

---

```
read-bookmark "elearning"
```

表示访问该网站。程序中目前不需要具体执行对该网站的访问动作，只需要将"elearning"这个网站的状态标记为已读。已读的网站在树形视图上用 "\*"作为标记。比如elearin标记为已读后，显示为 \*elearning

网站的访问状态信息只在工作空间内有效，切换文件后就丢弃掉这些信息，目前对保存访问状态不做要求。

## 注意：

网站的标签可能会追加多种信息。比如，

1. 如果用户希望显示阅读的次数，elearning网站标签可以显示为 \*elearning[3]。
2. 在文件的属性结构上，用不同的方式标记用户当前正在编辑的文件，以及用户打开过的文件

程序需要能够支持以一种灵活的，基于插件的方式增强标签的显示信息。

---

---

---