

Lab 1 Html编辑器设计

1 项目架构

```
html_editor/
├── app/
│   ├── HtmlEditorApp.java
│   └── HtmlEditorCommandParser.java
├── controller/
│   ├── HtmlController.java
│   └── events/
├── io/
│   └── HtmlIO.java
├── model/
│   ├── TreeNodeModel
│   └── HtmlModel/
├── service/
│   ├── HtmlService.java
│   ├── SpellCheckServic.java
│   └── IDManager.java
├── utils/
│   ├── command/
│   ├── factory/
│   ├── observer/
│   ├── strategy/
│   └── visitor/
├── view/
└── Main.java
```

层级说明

model层

储存了树形结构的接口以及由组合模式组成的HtmlElement composite leaf 具体的各种Html元素以及document

io层

使用jsoup读写html并转换为我的html模型

service层

包括管理id和html元素的服务，他们被用于HtmlDocument 包括拼写检查的服务模块

utils层

包含设计模式的内容 命令模式 用于各类具体命令 包含CommandManager用于执行命令以及一系列command
工厂模式 用于创建Html元素 分为抽象工厂（用于Treenode）以及具体工厂（用于HtmlElement） 观察者模式
用于同步model-controller-view之间的更新关系 策略模式 用于得到Html不同的转化格式(tree/indent) 访问者模式
用于访问HtmlDocument进行各种操作，如更新 读取文本用于检查等

view层

用于执行控制台打印

controller层

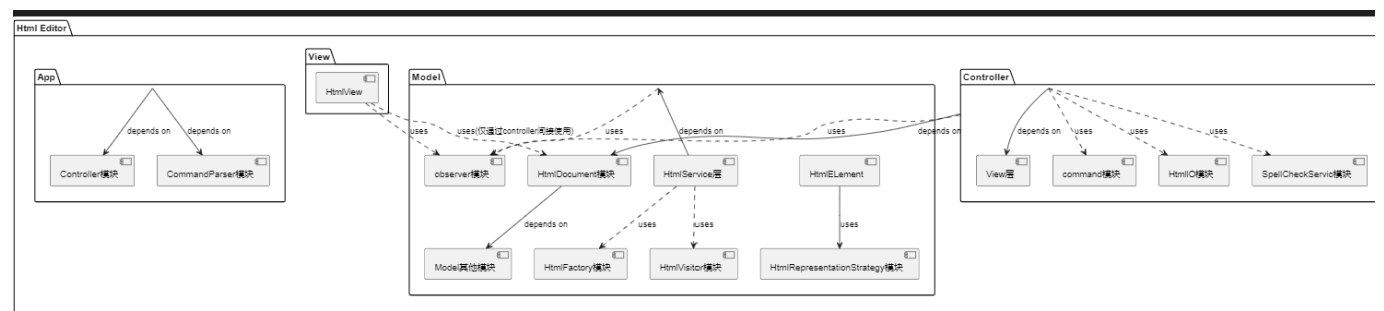
统合上述层 管理view、HtmlDocument(即model)以及拼写检查service、commandManager等服务

app层

创建主循环以及controller和CommandParser实例 由parser读取命令，交由controller进行所有操作

依赖关系

依赖关系如图



文字说明

1. Model 包 HtmlElement模块依赖于 HtmlRepresentationStrategy模块来转化为字符串 HtmlDocument模块依赖于 Model其他模块和HtmlService实现完整的文档功能 HtmlService层：依赖于整个 Model，管理

- HtmlElement的各种操作 使用 HtmlFactory模块 和 HtmlVisitor模块使用工厂模式创建新元素，使用访问者模式进行访问 Observer模块：Model 依赖于 observer模块，实现Observable功能
2. Controller 包 Controller依赖于 HtmlDocument模块 和 View层，协调命令执行、数据管理与显示。使用 command模块、observer模块、HtmlIO模块 和 SpellCheckService模块。观察Model和View，同时具备IO和拼写检查的功能。
 3. View 包 HtmlView间接使用 HtmlDocument模块（通过 Controller），通过 Controller 获取 HtmlDocument 的数据 使用 observer模块，实现Observable功能
 4. App 包 App依赖 Controller模块 和 CommandParser模块，是程序的入口。

测试

使用Junit进行测试，针对所有模块都有一个或多个测试。为了运行测试，输入以下命令

```
cd path/to/pom.xml
mvn test
```

特别注意：不要随意移动内部已经包含的文本文档

2 功能细节

文档要求的功能

init

```
[Console] init
-----
Current Document Structure:
-----

<html>
  <head>
    <title></title>
  </head>
  <body></body>
</html>
-----
[Console] Successfully init document
-----
[Console]
```

append

```
[Console] append li li_1 body
-----
Current Document Structure:
-----

<html>
  <head>
    <title></title>
  </head>
  <body>
    <li id="li_1"></li>
  </body>
</html>
-----
[Console] Successfully append element to body
-----
```

insert

```
[Console] insert li li_0 li_1
```

```
-----  
Current Document Structure:  
-----
```

```
<html>  
  <head>  
    <title></title>  
  </head>  
  <body>  
    <li id="li_0"></li>  
    <li id="li_1"></li>  
  </body>  
</html>
```

```
-----  
[Console] Successfully insert element before li_1  
-----
```

```
[Console]
```

delete

```
Current Document Structure:
```

```
-----  
<html>  
  <head>  
    <title></title>  
  </head>  
  <body>  
    <li id="li_2">helloworld</li>  
    <li id="li_1">Text</li>  
    <li id="li_3"></li>  
  </body>  
</html>
```

```
-----  
[Console] Successfully read document from 11.html  
-----
```

```
[Console] delete li_1  
-----
```

```
Current Document Structure:
```

```
-----  
<html>  
  <head>  
    <title></title>  
  </head>  
  <body>  
    <li id="li_2">helloworld</li>  
    <li id="li_3"></li>  
  </body>  
</html>
```

```
-----  
[Console] Successfully delete element of id li_1  
-----
```

```
[Console]
```

edit-id

```
[Console] edit-id li_0 li_2
```

```
-----  
Current Document Structure:  
-----
```

```
<html>  
  <head>  
    <title></title>  
  </head>  
  <body>  
    <li id="li_2"></li>  
    <li id="li_1"></li>  
  </body>  
</html>
```

```
-----  
[Console] Successfully edit id  
-----
```

```
[Console]
```

edit-text

```
[Console] edit-text li_1 Text
```

```
-----  
Current Document Structure:  
-----
```

```
<html>  
  <head>  
    <title></title>  
  </head>  
  <body>  
    <li id="li_2"></li>  
    <li id="li_1">Text</li>  
  </body>  
</html>
```

```
-----  
[Console] Successfully edit content of li_1  
-----
```

```
[Console]
```

print-tree

```
[Console] print-tree
```

```
-----  
Current Document Structure:  
-----
```

```
html #html  
├── head #head  
│   └── title #title  
└── body #body  
    ├── li #li_2  
    └── li #li_1  
        └── Text
```

```
-----  
[Console] Successfully converted to tree representation  
-----
```

```
[Console]
```

print-indent

```
-----
Current Document Structure:
-----

<html>
  <head>
    <title></title>
  </head>
  <body>
    <li id="li_2"></li>
    <li id="li_1">Text</li>
  </body>
</html>
-----

[Console] Successfully converted to indented representation with indent 2
-----
[Console]
```

spell-check

```
-----
Current Document Structure:
-----

<html>
  <head>
    <title></title>
  </head>
  <body>
    <li id="li_2">helloworld</li>
    <li id="li_1">Text</li>
  </body>
</html>
-----

[Console]
[Spell Check Result]

[Id: li_2]
[Error: Possible spelling mistake found.] [Type: Possible Typo] [Position 0-10] [Suggestions: hello world]
-----
```

undo

以append为例

```
[Console] append li li_3 body
```

```
Current Document Structure:
```

```
<html>
  <head>
    <title></title>
  </head>
  <body>
    <li id="li_2">helloworld</li>
    <li id="li_1">Text</li>
    <li id="li_3"></li>
  </body>
</html>
```

```
[Console] Successfully append element to body
```

```
[Console] undo
```

```
Current Document Structure:
```

```
<html>
  <head>
    <title></title>
  </head>
  <body>
    <li id="li_2">helloworld</li>
    <li id="li_1">Text</li>
  </body>
</html>
```

```
[Console] Successfully undo command append
```

```
[Console]
```

redo

以append为例

```
[Console] undo
```

```
Current Document Structure:
```

```
<html>
  <head>
    <title></title>
  </head>
  <body>
    <li id="li_2">helloworld</li>
    <li id="li_1">Text</li>
  </body>
</html>
```

```
[Console] Successfully undo command append
```

```
[Console] redo
```

```
Current Document Structure:
```

```
<html>
  <head>
    <title></title>
  </head>
  <body>
    <li id="li_2">helloworld</li>
    <li id="li_1">Text</li>
    <li id="li_3"></li>
  </body>
</html>
```

```
[Console] Successfully redo command append
```

```
[Console]
```

read

```
[Console] read 11.html
```

```
Current Document Structure:
```

```
<html>
  <head>
    <title></title>
  </head>
  <body>
    <li id="li_2">helloworld</li>
    <li id="li_1">Text</li>
    <li id="li_3"></li>
  </body>
</html>
```

```
[Console] Successfully read document from 11.html
```

```
[Console]
```

save


```
[Console] save 11.html
```

```
-----  
Current Document Structure:  
-----
```

```
<html>  
  <head>  
    <title></title>  
  </head>  
  <body>  
    <li id="li_2">helloworld</li>  
    <li id="li_1">Text</li>  
    <li id="li_3"></li>  
  </body>  
</html>
```

```
-----  
[Console] Successfully write document to file  
-----
```

```
[Console]
```

未知命令识别

```
-----  
[Console] appens  
[Console] Unknown command: appens  
[Console] The closest command is append  
[Console]
```

3 运行

项目管理

本项目使用Maven进行构建并管理依赖

jre版本: java 21

```
openjdk version "21.0.2" 2024-01-16 LTS  
OpenJDK Runtime Environment Temurin-21.0.2+13 (build 21.0.2+13-LTS)  
OpenJDK 64-Bit Server VM Temurin-21.0.2+13 (build 21.0.2+13-LTS, mixed mode, sharing)  
PS D:\Code\Java\2024 Fall Software Design\Lab_1\html_editor>
```

运行方式

直接使用java运行打包好的jar文件，已包含所有依赖的库

```
PS D:\Code\Java\2024 Fall Software Design\Lab_1\html_editor\target> java -jar html_editor.jar
```

```
-----  
HTML Editor  
Version 1.0.0  
Copyright: None  
-----
```

额外命令

除了要求的命令外，提供help命令打印所有可用的命令

```
[Console] help
-----
[Console] Available Commands:
[Console] init [filename] [title]
[Console] read filepath
[Console] save filepath
[Console] spell-check
[Console] insert tagName idValue brotherId [textContent]
[Console] edit-id oldId newId
[Console] redo
[Console] delete id
[Console] print-indent [indent]
[Console] undo
[Console] edit-text id [newTextContent]
[Console] print-tree
[Console] append tagName idValue parentId [textContent]
-----
[Console]
```