功能概述

Lab1是一个基于命令行的简易文本编辑器（文本在这里认为是一个长字符串），启动后从一个空白的字符串开始，也可以在命令行用`-t "init text"`在命令行设置初始文本（可选择任意合法字符串作为此处输入），用户可以输入命令对文本进行编辑和操作。
命令分为两类：

1. 非修改类命令
2. 修改类命令

其中，非修改类命令只是查看文本的状态信息而不会修改文本的内容，而修改类命令会导致当前编辑文本的内容改变。

修改类的命令需要支持undo/redo的操作。

只要执行修改类的命令，就需要在控制台输出当前最新文本内容。

命令列表(55%)

文本编辑部分

1. 显示当前编辑的字符串

`s`

2. 在尾部添加字符串（修改类命令）

`A "last word"`

命令运行后，”last word“（可选择任意合法字符串作为此处输入）添加到当前缓存中的字符串的尾部

3. 在头部添加字符串（修改类命令）

`a "first word"`

命令运行后，”first word“（可选择任意合法字符串作为此处输入）添加到当前缓存中字符串的头部

4. 从尾部删除指定数量的字符（修改类命令）

`D 5`

如果字符串长度n小于5（可选择任意合法数值作为此处输入），则删除n个字符

5. 从头部删除指定数量的字符（修改类命令）

`d 5`

如果字符串长度n小于5（可选择任意合法数值作为此处输入），则删除n个字符

6. 列出最近执行的修改类命令的列表

`l 10`

显示出最近执行的10（可选择任意合法数值作为此处输入）个命令及其序号，最近执行的排在前面，序号为1

7. undo操作

`u`

取消上一步操作（修改类命令），可连续取消直到所有操作（修改类命令）已全部被取消

8. redo操作

`r`

重做上一步undo取消的操作，可连续redo直到所有被undo取消的操作已全部被重做

9. 定义宏

`m 5 m10`

将最近执行的5（可选择任意合法数值作为此处输入）个修改类命令组成一个宏命令，名称为m10（可选择任意合法字符串作为此处输入）。程序将增加一个修改类命令，可以使用 `$m10` 运行该命令

拼写检查部分(35%)

10. 指定文本语言

默认语言为英语

- `lang eng` 指定当前编辑区中语言为英语
- `lang fra` 指定当前编辑区中语言为法语

11. 指定文本格式

默认为普通文本格式

- `content txt` 普通文本格式
- `content xml` xml格式，
  例如

```
<e><t>title</t><d>the Stream is actually being modified during access</d></e>
```

注意： xml在拼写检查时需要忽略tag的标签文字，比如上面的`e`,`t`,`d`等，只检查tag围定的文字段落的内容。

12. 执行拼写检查

`spell`

根据文本类型和语言，合理地选择相关模块，执行拼写检查，完成后列出所有未通过拼写检查的单词。

为减少程序的复杂度，对可进行拼写检查的文本做以下假设：

1. 文字段落中只包括ascii空格、逗号、句号和所有的大小写字母。空格、逗号、句号为分隔符。

2. 拼写检查可以使用自己定义的简单的词库，英文和法文词库分别定义在`eng.txt`和`fra.txt`中。

此外，需要设计能够方便扩展的程序架构，以便：

1. 支持不同的可拼写检查语言（可能是定义了不同的词库，也可以对接不同的拼写检查API）
2. 支持不同的格式，比如支持markdown格式，需要过滤其中的某些内容，比如link

Bonus (10%)

13. 标记拼写错误

`spell-a`

与`spell`命令基本相同，只是输出的是加上了错误标记的文本。比如：
编辑区域中的文本为

```
the Srteam is actually being modified during access
```

则会输出

```
the *[Srteam] is actually being modified during access
```

注意：**这一操作并不修改编辑区域中的文本**，需对所有格式的文本有效。

14. 删除错误单词（修改类命令）

`spell-m`
编辑区域中的文本原本为

```
the Srteam is actually being modified during access
```

运行后会变为

```
the  is actually being modified during access
```

注意：需对所有格式的文本有效。

测试用例

参见附件（附件会在2021.11.04前发布到elearning上）

评分标准

1. 每个命令至少有一个自动测试用例，需要全部通过，请在测试代码中标记测试的内容。 40%
2. 通过TA设计好的手动测试用例。 40%
3. 程序结构及相关设计模式的运用。 20%

语言

建议采用java或者C++，并且只使用标准库。

时间

截止日期：2021.12.04 23:59