

期中之前的任务: vim

任务描述:

Vim是从vi发展出来的一个文本编辑器。其代码补完、编译及错误跳转等方便编程的功能特别丰富, 在程序员中被广泛使用。和Emacs并列成为类Unix系统用户最喜欢的编辑器。

这个任务需要大家用到简单GUI,字符串的处理,一定的算法和数据结构,网络编程相关知识

功能要求:

- 一: (基本字符串操作)
 1. 在不同模式下,键入 `i` 进入输入模式, 此时可以正常编辑文本, `ESC` 进入普通模式, 此时不可编辑文本, 在普通模式下, 按 `:` 进入命令模式。
 2. 在普通模式下, 分别通过 `h`、`j`、`k`、`l`,进行光标的左、上、下、右移动(上下移动为难点)
- 二: (IO)
 1. 普通模式下, 键入 `:q` 进入命令模式并执行命令 `q`, 不保存退出(如果文件被修改, 则提示未保存, 并终止退出命令, 即只有文件未做修改是才能退出)。
 2. `:w` 保存但不退出。
 3. `:x` 保存并推出。
 4. `:q!` 文件已修改但不想保存修改, 使用则放弃修改并退出。
 5. `dd` 删除一行
 6. `yy` 复制一行
 7. `p` 将复制的内容粘贴到光标所在的行的后面。
- 三: (多线程, 网络操作、字符串高级操作,正则表达式)
 1. 定时备份文件。
 2. (尽力做) 远程编辑文件(这台电脑编辑另一台电脑)
 3. 普通模式下, 按 `/` 进行字符串匹配, (如 `/foo`, 则匹配文本中的 `/foo` 字符串)
 4. 普通模式下, 按 `:%s/foo/fool`, 查找 `/foo` 并替换为 `/fool` (示例 `!foo`、`fool` 均为用户输入)
- 四: (代码重构、宏录制、高级功能)
 1. `qa` 开始记录键盘操作, 并记录到寄存器 `a` 中。再按 `q` 完成记录。按 `@a` 重复 `a` 中记录的动作。
 2. 减少重复代码, 使用面向对象的设计方法,注重健壮性和交互性

任务提示&建议:

1. GUI不是重点,能用就行,可以不用很漂亮
2. 代码可以直接运行看到GUI界面,可以不用命令行再输入vim去启动界面
3. 打包格式和上一个任务的shell一样
4. 可以多看看抽象类和接口的知识,用面向对象的思想写出更优雅的代码