Руководство программиста

Назначение программы

См. ТЗ.

Функции, исполняемые программой

Принимает через командную строку 3 параметра. Проверяет корректность параметров:

* Файлы текста и словаря должны существовать, быть не пустыми и с правами на чтение.
* Параметр N размера html-файла должен быть числом в интервале [10..10000].

В случае успешной проверки выполняет обязательные требования к ТЗ и выводит время, понадобившееся для решения задачи, иначе выводит описание ошибки. **Если предложение содержит больше N строк возникнет ситуация, когда html-файл будет содержать больше N строк.**

Условия, необходимые для выполнения программы

Java SE version 8 update 11. ОС Windows 7 64bit. IDE Eclipse (Luna 4.4.0). У меня такая. На других платформах не тестировалось, но должно работать по идее. Для корректного отображения текста в html-файлах, кодировка файла текста и файла словаря должна быть “windows-1251”. Html-файлы сохраняются в директории, где лежит exe. Для корректного сохранения html необходимо права на запись файлов в данную директорию.

Вопросы, которые у меня возникали в процессе решения

Q1: Как java обрабатывает различные стандартные разделители файла? В Windows и Linux они разные.

A1: Метод readline() нормально обрабатывает и файлы, созданные в Linux, и файлы, созданные в Windows. При имеющихся ограничениях 2 mb переполнения строки быть не должно.

Q2: Как определить кодировку файла?

A2: Проблема сложная и в общем случае решения не имеет. Берется кодировку из настроек системы по умолчанию. В программе явно используется кодировка “windows-1251”. Определена в виде константы. Впоследствии можно вынести в параметры командной строки или определять программно. Можно прикрутить jChardet.

Q3: Что считать концом предложения?

A3: Считаем точку, восклицательный знак и вопросительный знак. Хотя есть еще UTF-символ многоточия и вообще предложение может быть с сокращениями и и т.д. Вынесем проверку символа на конец предложения в отдельную функцию.

Q4: Проверять ли формат словаря?

A4: В ТЗ указано, что словарь содержит ровно одно слово в строке. Значит так оно и есть. Можно сделать функции LoadFile булевской – и потом при чтении обрабатывать строку (смотреть, что слово там одно).

Q5: Какую коллекцию использовать для хранения словаря? Нам понадобится две операции: загрузка из файла, и проверка слова на вхождение в словарь.

A5: Лучше всего для этого подойдет АВЛ-дерево. TreeSet, так как там красно-черное и логарифмическая операция поиска. На всякий случай вынести коллекцию в отдельный класс, чтоб потом можно было легче сменить коллекцию, если потребуется.

Q6: Что использовать для разбиения строк?

A6: Нашел 2 метода: использовать StringTokenizer() или Split().StringTokenizer() , так как для данной задачи он производительней, хоть и считается устаревшим. В Split() есть регулярные выражения, но они здесь не особо нужны. На всякий случай вынести в отдельный класс, чтоб потом можно было легче сменить метод, если потребуется. Вынести разделители в отдельную константу.

Q7: Что считать верным вхождением в словарь? У символов строки может быть разные регистр “ЛеСеНкА” или словарное слово может быть только частью слова “оПУШКА” и “пушка” соответственно.

A7: Хранить словарные слова в верхнем регистре и при проверке переводить проверяемое слово в верхний регистр. Считать совпадением только полное совпадение слов в верхнем регистре. Использовать Equal(), так как == сравнивает ссылки. Посмотреть на EqualsIgnoreCase().

Q8: Какой метод использовать, чтобы выделять слова жирным и курсивом?

A8: Два варианта: либо связка тегов <b> и <i>, либо в теге < span class=”…” > . Второй вариант лучше, так как можно стиль отображения как угодно в одном месте настроить.

Q9: HTML обрезает подряд идущие пробелы. Что делать?

A9: Лечится использованием тега <Pre>.

Q10: В Java есть StringBuffer, String, StringBuilder и char[]. Какой лучше для данной задачи?

A10: String – immutable. Каждый раз создается экземпляр. Использовать только если какой-то метод уже выдает результат в String. char[] лучше использовать для паролей. StringBuffer и StringBuilder mutable. Но  StringBuffer  синхронизирован, а StringBuilder нет. StringBuilder шустрее. Многопоточности не требуется – так что StringBuilder.

Q11: Как обеспечить валидность html-файла?

A11: проверять до победного на <http://validator.w3.org/>.

Q12: Какой стиль оформления кода в Java?

A12: <http://habrahabr.ru/post/112042/>. Лень. Силы иссякли.

Q13: Как сформировать Jar-файл в Eclipse?

A13: File->Export->Runnable Jar.

Q14: Как сформировать exe-файл?

A14: Программа launch4j.

Комментарии

Требования к качеству выполнено кое-как, дополнительные требования тоже. Комментарии в коде есть – но не JavaDoc. Стиль форматирования и кодирования тоже не Java. Чую с ООП тоже не все гладко. Необходимый и достаточный результат для данной работы получен. Перфекционизмом не страдаю. Причины использования тех или иных вещей – в комментариях к коду и в прошлом разделе.