Разрешается делать любую из двух на выбор. Если есть желание или неуверенность, что сделал правильно - то можно и обе.

Задача 1. Создать программу обработки текста учебника по программированию с использованием классов: Символ, Слово, Предложение, Знак препинания и др. (состав и иерархию классов продумать самостоятельно). Во всех задачах с формированием текста заменять табуляции и последовательности пробелов одним пробелом.

Вывести все предложения заданного текста в порядке возрастания количества слов в каждом из них.

Во всех вопросительных предложениях текста найти и напечатать без повторений слова заданной длины.

Из текста удалить все слова заданной длины, начинающиеся на согласную букву.

В некотором предложении текста слова заданной длины заменить указанной подстрокой, длина которой может не совпадать с длиной слова.

--------------------------------------------

Задача 2. Обычной проблемой анализа текстов является определение частоты и расположения слов в документе. Эта информация запоминается в конкордансе, где различные слова перечислены в алфавитном порядке и каждое слово снабжено ссылками на строки текста, в которых это слово встречается. Рассмотрим следующую цитату.

Peter Piper picked a peck of pickled peppers. A peck of pickled

peppers Peter Piper picked. If Peter Piper picked a peck of

pickled peppers, where is the peck that Peter Piper picked?

Слово ""piper"" встречается здесь 4 раза в строках 1, 2 и 3. Слово ""pickled"" встречается 3 раза в строках 1 и 3. Результат обработки этого текста представляется списком слов в алфавитном порядке, для каждого слова указывается частота его появления в тексте (количество вхождений) и список номеров строк, в которых встречается это слово. Если слово встречается в строке более одного раза, то номер этой строки указывается в списке только один раз:

a.................................3: 1 2

if.................................1: 2

is................................1: 3

of................................3: 1 2

peck............................4: 1 2 3

peppers.......................3: 1 2 3

peter............................4: 1 2 3

picked..........................4: 1 2 3

pickled.........................3: 1 3

piper............................4: 1 2 3

that..............................1: 3

the...............................1: 3

where...........................1: 3

Требуется: осуществить обработку текстового файла, содержащего некоторый многостраничный текст. Будем считать, что одна страница текста состоит из N строк. Результ обработки – текстовый файл, содержащий предметный указатель встречающихся в тексте слов. В предметном указателе перечисляются слова в алфавитном порядке. Для каждого слова указывается количество его вхождений в текст и номера страниц, на которых встречатся это слово. При формировании предметного указателя, слова, начинающиеся с разных букв алфавита разделяются на группы:

А

a...............................120: 1 2 3 4 5 6 7 8 9 10 11 12

and............................89: 1 2 3 6 8 9 12

apple...........................3: 7 9

B

bag.............................2: 11 12

big..............................7: 5 7 10

C

cake..........................1: 3

circle..........................3: 8 9

. . .

и т.д."

1. Выполнить анализ и декомпозицию предметной области.
2. Использовать возможности ООП: наследование, полиморфизм, инкапсуляция. Отразить декомпозицию в структуре классов.
3. Каждый класс должен иметь исчерпывающее смысл название и информативный состав.
4. При кодировании следует придерживаться соглашения об оформлении кода code convention. (TO DO, FAQ)
5. Файлы проекта должны быть разделены по папкам согласно доменной модели.
6. Работа с консолью или консольное меню должно быть минимальным.

Использование интерфейсов и низкая связанность между классами;

Управление ресурсами и использование интерфейса IDisposable (TO DO);

Предпочтительное использование LINQ запросов при выполнении операций (TO DO);

Разумный exception handling;

Использование конфигурации для хранения путей к файлам;

Возможность сохранения объектной модели в виде текстового файла;

"