

Задача 11. Строки

А то как-то пропустили данную тему.

Заранее придумать не менее 5 различных тестов, охватывающих как типичные, так и все возможные граничные (наиболее невероятные и показательные) ситуации.

Как обычно, решение должно быть оформлено в виде отдельной функции / функций или класса.

Если в задаче используются регулярные выражения, то вы должны очень хорошо понимать, что это, и все особенности применительно к вашей задаче!

Если в вашем решении используется многократная конкатенация строк (не один-два-три раза, а именно многократная), то для эффективности использовать `StringBuilder`.

В задачах, где необходимо выбрать что-то без повторений, удобным (и эффективным в плане быстродействия) будет использовать `Set<String>` (`HashSet<String>` или `TreeSet<String>`) для хранения уже выбранных элементов. В задачах, где необходимо строке сопоставить какое-то значение, можно воспользоваться `Map<String, Integer>` (`HashMap<String, Integer>` или `TreeMap<String, Integer>`).

Варианты:

1. Реализовать функции кодирования и декодирования текста заменой букв текста симметричными относительно середины алфавита буквами, т.е. буква А должна заменяться буквой Я (а Я, наоборот, буквой А), Б – буквой Ю (а Ю – буквой Б) и т.д. Такая же логика действует для замены латинских букв. Меняться должны как прописные, так и строчные буквы (при этом прописные буквы остаются прописными, а строчные – строчными).
Подсказка: в программе в виде строковой константы должен быть задан русский и латинский алфавит и все манипуляции с текстом производятся работой с данной константой (ни в коем случае программа не должна содержать отдельные условные операторы для замены каждой буквы).
2. Реализовать функции кодирования и декодирования текста циклическим сдвигом букв алфавита. Величина сдвига передается вторым параметром в данные функции. Например, если величина сдвига равна 2, то все буква А меняется В, Б – на Г, В – на Д, ..., Ю – на А, Я – на Б. Такая же логика действует для замены латинских букв. Меняться должны как прописные, так и строчные буквы (при этом прописные буквы остаются прописными, а строчные – строчными).
Подсказка: в программе в виде строковой константы должен быть задан русский и латинский алфавит и все манипуляции с текстом производятся работой с данной константой (ни в коем случае программа не должна содержать отдельные условные операторы для замены каждой буквы).
3. Выбрать (в виде списка) из текста все слова без повторений, содержащие 3 и более одинаковые буквы. Разделителями слов считаются любые символы, отличные от букв А-Я, А-Z и цифр.
4. Обработать текст следующим образом:
 - несколько пробелов заменить на один;
 - убрать все пробелы перед знаками препинания (точка, !, ?, запятая, точка с запятой), а после данных знаков препинания должен быть пробел.

5. Удалить из текста все фрагменты в круглых скобках. Учесть, что скобки могут быть вложенными, например, из текста "Программа разработана на С# (С# (произносится си шарп) – объектно-ориентированный язык программирования, разработанный компанией Microsoft) студентами ВГУ (Воронежский государственный университет)!" должен получиться текст "Программа разработана на С# студентами ВГУ!".
6. (*) Выбрать (в виде списка) из текста все даты. За дату будем принимать фрагменты текста вида:
 - "ДД.ММ.ГГГГ" (Д, М, Г – цифры), например "20.12.2017";
 - "Д(Д) месяца ГГГГ", например, "7 ноября 1917";
 - "Д(Д) месяца" (без указания года, если год не указан), например "8 Марта" или "23 февраля".
7. (*) Выбрать из текста имена людей. За имя человека будем принимать фрагменты вида:
 - "Иванов Иван Иванович" (через пробел три слова подряд, состоящих только из русских букв, где первая буква – большая, остальные – малые);
 - "Алгазинов Э.К." (фамилия с большой буквы + инициалы, также большие буквы).Естественно, падежи не учитываем, т.е. могут встретиться имена в виде "Медведевым Д.А.", которые по описанным выше правилам также должны быть выбраны.
8. Обработать текст следующим образом: в конце каждого слова приписать в скобках длину данного слова. Словом считается непрерывная последовательность символов (строчных и прописных) А-Я, А-Z и цифр. Например, в результате обработки текста "Придумайте сложный пароль вида Az12-67\$AF2" должен получиться текст "Придумайте(10) сложный(7) пароль(6) вида(4) Az12(4)-67(2)\$AF2(3)".
9. Дан текст, состоящий из предложений. Концом предложения считаются символы точка, '!' и '?'. Началом предложения – любой символ, отличный от данных трех и пробела, первый после конца предыдущего предложения (или вообще первый в тексте). Выбрать (в виде списка) из текста все вопросительные предложения.
10. Обработать HTML-файл, убрав все теги, т.е. должен остаться только текст. Каждую последовательность удаленных тегов заменить на один пробел. Например: "<html><head><title>Моя страница</title><head><h1>Привет, Я Колян!
Закурить есть?</h1></head>/body>" после удаления тегов превратится в "Моя страница Привет, Я Колян! Закурить есть? "
11. Дан текст, состоящий из предложений (см. предыдущую задачу). Необходимо пронумеровать предложения в тексте, приписав номер предложения в конце предложения (после знака окончания предложения) в квадратных скобках. Пример: "Мама мыла раму. Вовочка валял дурака..." → "Мама мыла раму.[1] Вовочка валял дурака.[2].."
12. (*) Примитивная система исправления орфографических ошибок. Задается текст, потенциально содержащий орфографические ошибки, и набор корней слов (в виде массива или списка слов). Если какой-либо фрагмент текста, отличается от какого-либо корня одним символом, то заменить этот фрагмент на соответствующий корень слова. Пример ("Корова дает малако каторае пьют дэти", {"молоч", "коров", "апельсин", "кот", "дети"}) → "Корова дает малако каторае пьют дети".
13. (*) Написать функцию представления числа (до миллиона включительно) в виде строки, например 217045 → "Двести семнадцать тысяч сорок пять".

14. Исправить текст, убрав подряд идущие одинаковые слова, разделанные одним или несколькими пробелами (довольно часто встречающаяся ошибка при печати текста). Словом считается непрерывная последовательность символов (строчных и прописных) А-Я, А-Z и цифр. При удалении повторяющегося слова пробелы между повторяющимися словами также убрать.
15. Написать функции представления арабского числа римским и наоборот. Арабские числа представлять целочисленным типом данных, римские – строкой.
16. Обработать текст программы на языке С#, удалив из текста все комментарии (как однострочные, так и многострочные).
17. Выбрать (в виде списка) из текста все числа. Числа могут быть как целыми, так и вещественными (с разделителем точка или запятая).
18. Выбрать (в виде списка) из программы на языке С# все ключевые слова (список ключевых слов должен быть задан в виде массива или списка строк). Учесть, что для корректного выделения ключевых слов необходимо правильно (в рамках данной задачи) разбить программу на элементы (т.е. если в программе встречается идентификатор `break_flag`, это еще не означает, что в программе есть ключевое слово `break`). Для простоты будем считать, что комментариев в программе нет.
19. (*) Обработать текст программы на языке С#, заменив все фрагменты вида "переменная = переменная + 1" (с любым количеством пробелов вокруг символов '=' и '+') на фрагменты "переменная++".
20. Задача, обратная предыдущей.
21. Реализовать программу транслитерации текста на русском языке английскими символами согласно следующим правилам: <http://translit-online.ru/pasport.html>
22. Выбрать (в виде списка) без повторений из текста все слова заданной длины. Словом считается непрерывная последовательность символов (строчных и прописных) А-Я, А-Z и цифр.
23. Выбрать (в виде списка) без повторений из текста все слова, в которых количество гласных букв больше количества согласных. Словом считается непрерывная последовательность символов (строчных и прописных) А-Я.
24. Закодировать текст следующим образом: в каждом слове переставить буквы в обратном порядке. Словом считается непрерывная последовательность символов (строчных и прописных) А-Я, А-Z и цифр. Например, "Помимо С# вы будете изучать Java, С++ и другие языки программирования." после кодирования превратится в "омимоП С# ыв етедуб ьтачузи аvaJ, С++ и ейгурд икызя яинавориммаргорп."
25. (*) Дан текст, состоящий из абзацев. Каждый абзац записан с новой строки. Отформатировать данный текст, выровнять текст по ширине, которая задается. Каждый абзац начинать с красной строки в 6 пробелов. Длинные абзацы переносить на новую строку. В строке текст выравнивать по ширине добавлением пробелов между словами. Результат (при открытии текста в блокноте) должен получиться примерно таким <http://lib.ru/RUFANT/BELAEW/doul.txt> (только без специального оформления названий разделов).

26. (*) Дата описывает составным типом данных (число, месяц, год). Реализовать функцию представления даты в виде строки на русском языке по шаблону. Должны поддерживаться, как минимум шаблоны YYYY, YY, MM, MON, MONTH, D, DD, DAY, DY (суть этих шаблонов см. http://oracleplsql.ru/to_char-function.html). Для шаблонов, которые предполагают название дня недели или месяца, должны учитываться большие и малые буквы шаблона (варианты: все малые, все большие, первая большая).
Пример: ({ Day: 1, Month: 9, Year: 2017 }, "Month DD, YYYY")) → "Сентябрь 01, 2017"
27. Выбрать (в виде списка) из текста все аббревиатуры. Аббревиатурой будем считать слово от 2 до 6 символов, состоящее только из прописных (больших) букв без чисел. Словом считается непрерывная последовательность символов (строчных и прописных) А-Я, А-Z и цифр.
28. Выбрать (в виде списка) из текста все слова, в которых встречаются, как русские, так и латинские буквы (типичная ошибка, когда вместо русской буквы печатается латинская с тем же начертанием, особенно с/с – расположены на одной кнопке русскоязычной клавиатуры). Словом считается непрерывная последовательность символов (строчных и прописных) А-Я, А-Z и цифр.
29. Выбрать (в виде списка) из текста все слова (без повторений), которые являются именами собственными. Такими будем считать слова, которые расположены не в начале предложения, но начинаются с заглавной буквы (остальные – строчные). Словом считается непрерывная последовательность символов (строчных и прописных) А-Я, А-Z и цифр. Концом предложения считаются символы точка, '!' и '?'. Началом предложения – любой символ, отличный от данных трех и пробела, первый после конца предыдущего предложения (или вообще первый в тексте).
30. (*) Из текста выбрать (в виде списка) без повторений все email-адреса. За email-адрес будет принимать любой фрагмент вида X@X, где X – любая последовательность букв английского алфавита, цифр, а также символов точка, '-' и '_' (естественно, первым и последним символом не может быть точка). При желании можно более полно проверять email-адрес согласно RFC 5322. При реализации регулярные выражения не использовать (как и для всех остальных задач, но для этой задачи стоит напомнить об этом еще раз отдельно).
31. Выбрать (в виде списка) из текста все предложения, в которых есть повторяющиеся слова. Учесть, что регистр букв в одинаковых словах может отличаться. Словом считается непрерывная последовательность символов (строчных и прописных) А-Я, А-Z и цифр. Концом предложения считаются символы точка, '!' и '?'. Началом предложения – любой символ, отличный от данных трех и пробела, первый после конца предыдущего предложения (или вообще первый в тексте).
32. Выбрать из текста самое длинное предложение. Если в тексте несколько предложений одинаковой длины – выбрать самое первое. Длина предложения считается по количеству слов. Словом считается непрерывная последовательность символов (строчных и прописных) А-Я, А-Z и цифр. Концом предложения считаются символы точка, '!' и '?'. Началом предложения – любой символ, отличный от данных трех и пробела, первый после конца предыдущего предложения (или вообще первый в тексте).