

Обработка строковых данных

Технические требования

- 1. Приложение должно начинать работу со считывания исходных данных из текстового файла, который может содержать один или несколько тестовых примеров. После этого пользователю предлагается ввести с клавиатуры свой тестовый пример.*
 - 2. В консольных вариантах приложений должна быть предусмотрена задержка консольного окна для просмотра результатов.*
1. Дана строка. Напечатать входящие в нее слова, но в обратном порядке (сначала последнее, потом предпоследнее и т.д.).
 2. Записать метод, возвращающий сумму чисел, встречающихся в заданной строке.
 3. Записать метод, возвращающий число вхождений подстроки в строку. Строка и подстрока являются неизменяемыми параметрами метода.
 4. Пусть даны две строки `str1` и `str2`. Необходимо выяснить, можно ли из `str1` путём перестановки литер получить строку `str2`. Записать метод, решающий указанную задачу.
 5. Записать методы `Encode` (зашифровать) и `Decode` (расшифровать), которые получают два параметра `str` и `alpha` строкового типа. В первом параметре задается слово, подлежащее шифрованию (расшифровке), второй представляет собой некоторую перестановку 26 латинских букв алфавита. Принцип преобразования для шифрации состоит в следующем. Если некоторая буква в `str` является *k*-ой буквой в обычном алфавите, то вместо нее должна быть взята буква из *k*-ой позиции "нового" алфавита `alpha`. Для метода дешифровки используется обратный принцип.
 6. Написать метод, который будет проверять, является ли строка правильным идентификатором, удовлетворяющим следующим требованиям: имя может содержать от 1 до 127 символов; первым символом не должна быть цифра; символами идентификатора могут быть буквы (строчные или прописные), цифры (от 0 до 9) и символ подчеркивания. Если обнаружена ошибка, необходимо выдать сообщение, квалифицирующее ее.
 7. Реализовать метод `IsTemplate` проверки соответствия строки шаблону. Шаблон – строка, в которой символы '?' и '*' интерпретируется специальным образом. Проверка соответствия выполняется по правилу:

 символу '?' должен соответствовать один произвольный символ в сравниваемой строке

 символу '*' должна соответствовать последовательность из нуля или более произвольных символов в сравниваемой строке

 любому другому символу должен соответствовать такой же символ в сравниваемой строке

Пример

```
IsTemplate("aaabbtt", "aa?b*t") -> True
IsTemplate("", "*") -> True
IsTemplate("", "?") -> False
IsTemplate("aaa", "aaa") -> True
IsTemplate("aaa", "aa") -> False
```

```
IsTemplate("aaa", "?aa") -> True  
IsTemplate("abaaaa", "??b*") -> False  
IsTemplate("abaaaa", "??*****") -> True
```

Проверяемая строка и шаблон являются параметрами метода.

8. Написать метод перевода числа до миллиона в текстовое представление. Пример:

34678 -> "тридцать четыре тысячи шестьсот семьдесят восемь"

9. Написать метод:

```
string[] split(string str, string pattern, out int count)
```

Данный метод должен разделять строку str на части и возвращать данные части в виде массива строк. pattern – строка, состоящая из одного или нескольких символов, каждый из которых рассматривается в качестве отдельного разделителя. Через count возвращается количество выделенных частей.

10. Реализовать метод, который распечатает в порядке встречаемости все слова некоторого текста, начинающиеся с заданной буквы. При этом каждое слово должно быть выведено ровно один раз
11. Реализовать метод, определяющий количество слов в тексте, у которых первый и последний символы совпадают. При этом каждое слово должно быть выведено ровно один раз.
12. Реализовать метод, который находит в тексте все слова максимальной длины.
13. Реализовать метод, который выводит каждое предложение в заданном тексте.
14. Реализовать метод, производящий вычисление значения целочисленного арифметического выражения, в котором использованы только операции сложения, вычитания и умножения..
15. Записать метод, проверяющий правильность расстановки скобок в алгебраическом выражении.
16. Реализовать метод, который в заданном тексте находит самый часто встречающийся алфавитно-цифровой символ. Вывести на экран символ и число его повторений.
17. Слова в строке разделяются пробелами. Зашифровать текст таким образом, чтобы каждое слово текста было записано в обратном порядке.
18. Даны две строки. Требуется написать метод, определяющий долю слов в первой строке, которая встречается и во второй строке («антиплагиат»).
19. Дан текст программы на языке C++. Требуется удалить из него все комментарии.
20. Реализовать метод, возвращающий массив слов длины N из заданной строки (знаки препинания, расположенные в начале и в конце слов, не учитывать). Если строка не содержит слов длины N, оставить результирующий массив пустым.