

Практика № 3

Программа с использованием классов объектного ввода-вывода и класса `string`

1. Формулировка задания

Определите класс для обработки строковой информации. Компонентой класса является объект класса **`string`** Стандартной библиотеки классов C++. Компонентная строка состоит из **слов**, разделителем слов является один или несколько пробелов. В начале и в конце строки может содержаться ноль или более пробелов.

Класс должен содержать конструктор по умолчанию, конструктор с параметром, конструктор копирования, перегруженную операцию присваивания, перегруженную операцию вывода содержимого объекта в поток.

Методы разрабатываемого класса должны обеспечить реализацию следующего функционала:

- чтение исходной строки из текстового файла, имя которого передается из командной строки. Для создания файла нужно воспользоваться стандартными средствами операционной системы (например, программой Notepad);
- формирование массива слов из исходной строки;
- обработку полученного массива в соответствии с вариантом задания;
- запись в результирующий файл **`result.txt`** результатов реализации алгоритма, предусмотренного вариантом задания.

Операции чтения из файла и записи в файл, а также консольный ввод-вывод должны быть реализованы средствами **объектного** ввода-вывода.

Основная программа (функция **`main`**) должна подтверждать работоспособность класса, содержать определение объектов класса различными способами и операторы, демонстрирующие вызов для этих объектов разработанных компонентных функций и перегруженных операций в соответствии с вариантами задания.

2. Варианты заданий разрабатываемой программы:

Вариант 1

Разработать метод, записывающий в результирующий файл из исходной строки слова, содержащие определенное количество символов. "Длина" слова вводится с клавиатуры.

Вариант 2

Разработать метод, который проверяет, являются ли слова, из которых состоит исходная строка, правильными идентификаторами языка C++. Правильные идентификаторы записать в результирующий файл.

Вариант 3

Разработать метод, который в исходной строке определяет самое короткое и самое длинное слова, и выводит их на экран и результирующий файл.

Вариант 4

Разработать метод, который проверяет, являются ли слова, из которых состоит исходная строка целыми десятичными константами C++. Правильные десятичные константы записать в результирующий файл.

Вариант 5

Разработать метод, который позволяет выбрать из исходной строки слова, начинающиеся и оканчивающиеся одним и тем же символом, выводит на экран и в результирующий файл сначала выбранные слова, а затем – все остальные.

Вариант 6

Разработать метод, который позволяет в исходной строке определить слова, начинающиеся с определенного символа, вывести их на экран и в результирующий файл (символ вводится с клавиатуры).

Вариант 7

Разработать метод, который позволяет вывести на экран и в результирующий файл из исходной строки слова, которые содержат не менее N вхождений некоторого символа. Количество вхождений N и изображение искомого символа вводятся с клавиатуры.

Вариант 8

Разработать метод, который позволяет упорядочить слова исходной строки по возрастанию их длины. В результирующий файл записать строку в упорядоченном виде.

Вариант 9

Разработать метод, который проверяет, являются ли слова, входящие в состав исходной строки, вещественными константами с фиксированной точкой. Выявленные вещественные константы записать результирующий файл.

Вариант 10

Разработать метод, определяющий в исходной строке самое длинное слово и выполняющий его реверс (первая буква меняется местами с последней, вторая – с предпоследней и т.д.). Полученное слово вывести в результирующий файл.

Вариант 11

Разработать метод, который позволяет подсчитать общее количество символов, содержащееся в нечетных словах исходной строки, вывести нечетные слова и количество символов в них на экран и в результирующий файл.

Вариант 12

Разработать метод, который выводит на экран и в результирующий файл слова, являющиеся положительными шестнадцатеричными константами C++.

Вариант 13

Разработать метод, который выводит на экран и в результирующий файл слова, начинающиеся с символов букв по убыванию их длины.

Вариант 14

Разработать метод, преобразующий слово, номер которого введен с клавиатуры (нумерация слов в строке начинается с единицы). Алгоритм преобразования: поменять в выбранном слове регистр символов. Исходное и преобразованное слова вывести в результирующий файл.

Вариант 15

Разработать метод, удаляющий из исходной строки слова длина которых не превышает некоторого значения R , вводимого с клавиатуры. Полученную в результате строку записать в результирующий файл.

Вариант 16

Разработать метод, который позволяет отсортировать слова исходной строки по убыванию их длины, используя для этого алгоритм сортировки вставками. Отсортированную последовательность слов записать в результирующий файл.

Вариант 17

Разработать метод, проверяющий имеются ли в исходной строке отдельные слова, заключенные в скобки (), [], { }. В результирующий файл вывести слово или слова, содержащиеся в скобках с ненарушенным балансом парности. Вложенность скобок отсутствует.

Вариант 18

Разработать метод, который подсчитывает количество слов, в которых встречается введенная с клавиатуры подстрока. Полученное число вывести на экран, а слова, в которых встречается данная подстрока – в результирующий файл.

Вариант 19

Разработать метод, рассчитывающий среднее количество символов в словах исходной строки. В результирующий файл вывести слова, длина которых превышает полученное среднее значение.

Вариант 20

Разработать метод, отыскивающий в исходной строке слова, являющиеся целыми положительными десятичными числами. Слова, являющиеся целыми десятичными числами преобразовать в числа и вывести в результирующий файл их сумму.

Вариант 21

Разработать метод, переворачивающий слова в предложении. Например, "Мама мыла раму" – "амаМ алым умар". Полученную строку вывести в результирующий файл.

Вариант 22

Разработать метод, формирующий аббревиатуру исходной строки. Например, "Федеральная служба охраны" – "ФСО". Исходную строку (пословно) и соответствующую ей аббревиатуру вывести в результирующий файл.

Вариант 23

Разработать метод, который объединяет в единое слово те слова из исходной строки, номера которых введены с клавиатуры (нумерация слов в строке начинается с единицы). Полученное слово вывести в результирующий файл.

Вариант 24

Разработать метод, позволяющий в исходной строке определить слова, заканчивающиеся символом цифры, вывести их на экран и в результирующий файл.

Вариант 25

Разработать метод, который выведет в результирующий файл слово (или слова) исходной строки наименьшей длины.