

Тестовое задание C++ программиста

Необходимо написать на чистом C++ класс, умеющий максимально быстро читать текстовые лог-файлы огромных размеров (сотни мегабайт, десятки гигабайт) и выдавать строки, удовлетворяющие условиям простейшего регекср: (как минимум операторы * и ?, более широкий набор возможностей приветствуется):

- символ '*' - последовательность любых символов неограниченной длины;
- символ '?' - один любой символ;
- должны корректно обрабатываться маски: *Some*, *Some, Some*, *****Some*** - нет никаких ограничений на положение * в маске.

Результатом поиска должны быть строки, удовлетворяющие маске.

Например:

1. Маска *abc* отбирает все строки содержащие abc и начинающиеся и заканчивающиеся любой последовательностью символов.
2. Маска abc* отбирает все строки начинающиеся с abc и заканчивающиеся любой последовательностью символов.
3. Маска abc? отбирает все строки начинающиеся с abc и заканчивающиеся любым дополнительным символом.
4. Маска abc отбирает все строки которые равны этой маске.

Класс не будет расширяться и не будет базовым классом. Единственная задача: сканирование и поиск текстовых ANSI (не unicode) строк в файлах.

Класс должен иметь следующий публичный интерфейс:

```
class CLogReader
{
public:
    CLogReader(...);
    ~CLogReader(...);
    bool  Open(...);           // открытие файла, false - ошибка
    void  Close();             // закрытие файла
    bool  SetFilter(const char *filter); // установка фильтра строк, false - ошибка
    bool  GetNextLine(char *buf, // запрос очередной найденной строки,
                      const int bufsize); // buf - буфер, bufsize - максимальная длина
                                         // false - конец файла или ошибка
};
```

Требование к реализации:

- Платформа: Windows XP и выше;
- Функциональность: максимальная производительность (кэширование результатов поиска и файла не требуется);
- Ресурсы: затраты памяти должны быть минимальны (в разумных пределах);
- Компоненты: запрещено использование сторонних библиотек (в том числе STL) и компонентов. Разрешено использовать только WinAPI и CRT;
- Исключения: нельзя использовать Windows и C++ исключения;
- Код должен быть абсолютно «неубиваемый» и защищённым от ошибок.

Требования к оформлению:

- Код должен быть максимально простым;
- Код должен быть максимально чистым, красивым и понятным;
- Результатом должен быть готовый класс (cpp + h файлы) и консольный пример (открыть текстовый файл в пару мегабайт и сделать выборку "order*closed") его использования. Файл и строка выбора должна указываться в аргументах примера;
- Результатом должен быть образец работающего exe файла + проект для MS Visual Studio 2005 и выше. (все в одном архиве).

Любые комментарии к коду приветствуются.