

Тестовое задание на позицию инженер-программист на C++

Messages queue

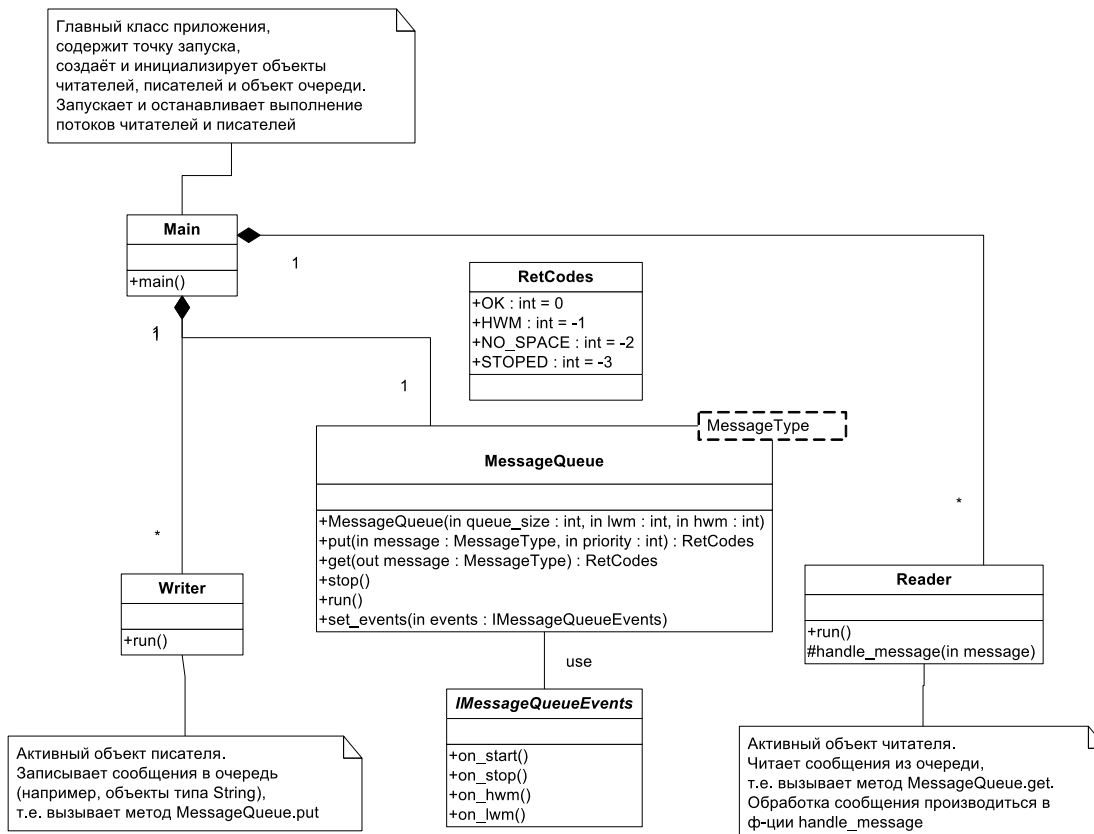
Задание:

Требуется реализовать блокирующую очередь с приоритетами, для асинхронного обмена сообщениями между несколькими потоками «писателей» и «читателей», и тестовое приложение, демонстрирующее его функциональность.

Класс MessageQueue должен поддерживать механизм HWM/LWM (high water mark/low water mark), приоритеты сообщений, и блокирующий вызов get (вызов блокируется, до появления сообщения в очереди).

Структура классов:

Общая схема:



Механизм нотификаций on_lwm/on_hwm обеспечивает возможность приостановки и возобновления деятельности писателей, в случае, когда очередь переполнена, и, соответственно, вновь свободна. Обычно, границы hwm и lwm выставляют 90% и 10% от размера очереди сообщений, соответственно.

Класс Writer:

Класс Reader:

Объект класса Reader находится в постоянном ожидании сообщений от MessageQueue. Остановится поток читателя только тогда, когда метод MessageQueue.get вернёт STOPPED.

Ф-ция handle_message вызывается для обработки каждого полученного сообщения, например, выводит сообщение на консоль или в файл.

Main:

Создаёт, инициализирует и запускает все объекты приложения. Затем ждёт некоторое время (какого либо сообщения от пользователя), останавливает очередь и дожидается корректного завершения всех потоков читателей и писателей, после этого останавливается.

Требуемый результат

В качестве результата, мы должны получить скомпилированный файл приложения, демонстрирующего работу класса MessageQueue (message_queue[.exe]), все исходные коды. И подробное описание всех функциональных методов всех реализованных классов, с указанием динамики поведения и требований потокобезопасности.

Notes:

- в приложении должно быть не менее двух объектов Writer;
- всё, что не описано в задании, трактуется по усмотрению кандидата;
- полученное приложение должно быть скомпилировано либо на Win32, либо на Solaris/SPARC, либо на Linux (x86 или x64).