

Modellazione di un processo consumatore per servire dei task a priorità diversa

Scrivere, in un linguaggio a scelta tra C#, Java o pseudocodice, l'algoritmo eseguito da un processo consumatore di richieste prodotte a ciclo continuo da altri processi.

Ogni richiesta è caratterizzata da una priorità, che può essere alta o bassa. L'algoritmo deve servire le richieste rispettando l'ordine di arrivo dando però precedenza alle richieste ad alta priorità.

Tuttavia, l'algoritmo deve anche evitare la starvation di quelle a bassa priorità, garantendo che il rapporto tra richieste servite ad alta priorità e quelle a bassa priorità sia al massimo di 4:1.

Si supponga di poter utilizzare la classe Queue (di cui non è richiesta implementazione), che implementa la seguente interfaccia:

```
/// <summary>
/// Represents a FIFO queue
/// </summary>
public interface IQueue
{
    /// <summary>
    /// Pushes the given request into the queue
    /// </summary>
    void Push(IRequest request);

    /// <summary>
    /// Pops the first request from the queue
    /// </summary>
    IRequest Pop();

    /// <summary>
    /// Sorts the requests contained in the queue by priority ascending.
    /// That is, low priority requests first, then high priority ones.
    /// </summary>
    void SortByPriority();
}

public interface IRequest
{
    Priority Priority { get; }
    void Process();
}

public enum Priority
{
    Low,
    High
}
```