

Priority Queue

The test below is meant to check a basic understanding of requirements analysis, data structures and programming principles. If you have this basic knowledge, producing good solutions will not take more than a few hours.

1 Description

Familiarity with basic FIFO queues is assumed:

[https://en.wikipedia.org/wiki/Queue_\(abstract_data_type\)](https://en.wikipedia.org/wiki/Queue_(abstract_data_type))

Implement a more abstract version of the queue that returns elements in order of their user-defined priority.

Do not use any existing implementation in the programming language you write this in as the idea is to build your own from the ground up.

Example Application: Accident and Emergency Reception in a Hospital

People arrive with various medical problems and wait to be seen by the doctors. The nurse assigns the priority based on the criticality of the patient. (*gunshot wound* > *broken leg* > *paper cut*). Priorities can change over time and the nurse will periodically review and update them and will also do that based on events (a person suddenly collapses and stops breathing). There is only one set of doctors and they treat one patient at a time. When they need to select a new person they just ask the nurse who is next. The nurse effectively manages a priority queue.

2 Implementation Requirements

- **pure Java** (no external libraries, no frameworks), **only** Java built-in data structures used
- the code structure, naming convention and style should be clean and easy to understand
- the core implementation must be as generic as possible, to be used as a library in various applications
- the core implementation needs to be basic as per the above use case (max 4 functional methods), about *100-150 lines* of code (spacing included)
- the implementation needs to be **tested**, by this surrounded by approx. *200-300 lines* of test code (spacing included)
- the implementation needs to support **multithreaded usage** and handle **race conditions**
- the implementation optionally can contain an implementation where the usage of the priority queue is demonstrated, however this is not needed
- the space and time performance of all operations should be optimized and discussed in code comments (https://en.wikipedia.org/wiki/Asymptotic_computational_complexity)
- the code needs to be published on a public Git repository (BitBucket or GitHub).

3 Restrictions

- You must complete the test within **48 hours** of receiving it.
- You can ask a set of questions after you reviewed this document, any questions about the requirements will be explained over chat, email or call.