PRATICAL ASSIGNMENT 2
of
Algorithms and Data Structures

INSTITUTO POLITÉCNICO DE BRAGANÇA
Escola Superior de Tecnologia e Gestão

Informatics Engineering / Management Informatics
May, 22th, 2018

Delivery deadline: 05/06/2018

DYNAMIC PRIORITY QUEUE IMPLEMENTATION OF LINEAR COMPLEXITY

The priority queue is a major ADT that does not follow the scheme of insertion and removal known as FIFO, typical of a queue. In it, the elements are removed by order of priority, which means that in the moment of removing, we select the highest priority element instead of the last one that was inserted. The order of priorities is established by the intrinsic order of collected elements themselves, being the most minor of them the highest priority element.

We intended, with this second assignment, implementing a priority queue of linear complexity, named by LinkedPriorityQueue, which should support one of the linked lists developed in the class. The priority queue should be an iterable collection of generic type elements. After the complete definition, build the main method that tests it properly, instantiating it and calling upon all his methods. In that test, it will be valued the collecting of objects of a class created by you for this purpose.

Work to develop

Implement in Java a solution for the problem above, that meets in full the specifications described next.

Considerations to be consider in the implementation

- The application must be developed in NetBeans IDE, JDK 1.7 or later, and the project should have the name "tp2_n1_n2", where n1 and n2 are the student's numbers of the two students of the group.
- The LinkedPriorityQueue class to develop must implement all the behavior defined in the PriorityQueue interface provided and be based on one of the linked list developed in the class.
- The PriorityQueue interface is available on ipb.virtual, subfolder <Recursos/avaliacao/TrabPrat2> area <AED> (along with the PriorityQueue.java source file, you can also find there the respective javadoc documentation, in MHTML and PDF format).
- It's forbidden to change anything from the PriorityQueue interface. Only methods that you cannot implement should pass comments on the interface (//...).
- In addition to commenting on methods that do not implement, you should mention them in the Javadoc block preceding the LinkedPriorityQueue class.
- Also, in that Javadoc block (which precedes the LinkedPriorityQueue Class) must include tags @author with the proper identification of the two elements of the group (name, number, and course).
- In addition to the javadoc block above, you don't need to include any other javadoc comments.
- The LinkedPriorityQueue class and the interface PriorityQueue should be in a package called "myqueues", the linked lists to support in a subpackage with the name "myqueues.linkedlists". The class with the main method and any other classes, which will be used for testing, should be placed on a second package, called "tests".

Global Rules (mandatory)

1. The assignment must be carried out by groups of 2 or 3 students.
2. This and the previous assignment are optional for students who have already achieved approval in the practical component of the curricular unit in one of the last two academic years. If the student choose not to perform both assignments, the classification in practice will be the same that the student obtain in the practical component in 2016/2017 or, in the absence of this, in 2015/2016.
3. Only shall be accepted for evaluation, the assignments whose implementation does not present any compilation error and with a minimum of features perfectly operational.

4. It's expressly forbidden to copy all or part of code from sources other than the documentation made available by the teachers of ADS.
5. The work must be delivered, within the delivery deadline, to the e-learning portal (at http://virtual.ipb.pt/, choose "Trabalho Pratico 2" in the "Trabalhos" tab, within the AED area).
6. It should be submitted the compressed folder of the project. To do this, from the File menu of NetBeans, select Export Project->To ZIP.
7. The assignment may only be submitted with a maximum delay of 5 days. The classification is subtracted by one, for each day of delay in the delivery of the assignment.
8. Resubmissions will not be allowed (when submitting ensure that it is the final version).
9. The students may need to present in person the work, on a date to scheduler by the teachers, demonstrating being able to implement the code, understand it and explain it.