

80007 Advanced Java Assignment 1: Keyword Snippet Analysis

Due: 11 pm **24th Sept 2019** (week 7) individual, worth 20%
plus demonstrations during lab 6 (10th Sep) and lab 7 (24th Sept).

Introduction

The kind of assignments we are seeking is the one that has significant scope for the application of the subject's content. This is quite a challenge since we need a problem that has potential for the nice application of the java IO, data structures, collections, documentation and the computational strength to make performance a significant issue in the quality of the solution.

As a result, we have opted for a keyword snippet analysis problem. The system must be implemented in Java.

Logistics

This is an individual assignment.

The assignment is worth a total of 20% of the total assessment.

The assignment is due by 11 pm, Tuesday, 24th Sep, 2019.

The assessment of the assignment will be based on the presence of working features and on submission of the assignment according to the rules and deadlines. If your assignment is found to contain the same or similar codes to another source and this has not been acknowledged, the marks penalty will be extremely high. Important: the portions of imported code from any source must be acknowledged as comments in the code by clearly delimiting the boundaries of the imported code and providing a reference to the source of the code. The readme.txt file should also indicate that there is imported code.

The submission of the assignment will be in electronic form. All assignment files will need to be packaged into a SINGLE zip (such as **Your-ID-FirstName-UG/PG-19S2-AJ-Assign1.Zip**) file and submitted to the Assignment 1 link in the Canvas. If you are undergraduate student, please mention UG, otherwise PG. Your zip file should contain a text file named "readme.txt" which should be created with a text editor such as notepad++, not a word processor. This file will contain identification information (name, id, and subject id); instructions on how to run the assignment, a list of known bugs, the identification of help you have received from outside the group including code that you have obtained from other sources, an itemization of the assessment items completed with an estimate of what you think you will get for the assignment and any other discussion that is needed. Note that the way you can receive feedback on your progress in the subject is through your progress in the assignment; you should be able to

have a fair idea of the mark you will get by what you have completed. You can also receive feedback and comments from your tutor.

Please note that this form of submission is not yet specified in the Assessment Details and Regulations.

Occasionally there will be problems with the submitted zip files; in such cases it is generally obvious whether the student has made a genuine attempt at the assignment and the student will be emailed and asked to resubmit the assignment. All students therefore need to retain a complete electronic copy of their work and maintain an up-to-date email address until they have received an assignment mark.

As an added precaution, check that it is possible to open the uploaded zip file from the Canvas.

A few days after the assignment deadline, the assignments will be marked; and it is expected that it will take at least 2 weeks after the deadline before the assignments will be marked.

The Problem

Keyword search systems have become an integral part of the data centric applications. The contemporary keyword search systems exploits the underlying data source statistics to recommend interesting and relevant content to the users.

The problem takes a list of keywords and a text file that contains pure text. We would like to exploit the so-called New I/O (or NIO) buffers and channels to read the text file content. Then, we copy the sections of the file that contains the defined keywords into the target text file. In fact, we extract the text snippets that contain the keywords of the keyword list. Include the following methods in your solution:

- FileChannel with direct ByteBuffer
- FileChannel with indirect ByteBuffer
- Buffered Stream I/O
- Programmer-managed Byte-Array

The response time for all methods should be recorded in a separate text file. You should use a batch file to include the java commands and command line arguments (to run the experiments for each method) in addition to the Eclipse. The experiments can be done by varying the buffer size and the text file which is supplied to the java program. There can be 3 command line arguments for setting up the experiments.

For example: *java programName methodName bufferSize textFileName*.

Also, build an index by using data structures such as hash tables to store each keyword (as the key) with its associated snippets (as the values for the key).

The total mark for the Assignment 1 will be obtained by adding all the marks, dividing by 10, rounding up to the next whole number and then subtracting marks for acknowledged code > 10% of total, any penalties for lateness and the inclusion of unacknowledged code.

The order of the feature list is also designed as a suggestion for the order of development and will guide the order of presentation of the lecture material. Ideally, you should have all the theory before you start the assignment, but this is, of course, not possible. Therefore, we have sequenced the lecture material in order to support the development of the assignment as much as is practicable.

Keyword Snippet Analysis (200 marks)

1. Design and implementation of different methods. (40 marks)
2. Use of batch file containing java commands to do the experiments. (10 marks)
3. Use NIO and advanced java features as far as possible. (40 marks)
4. Use advanced data structure for indexing and exception handling. (40 marks)
5. Recording the time for each method and printing the results in a text file. (50 marks)
6. Include reasonable documentation according to the Javadoc standards. (10 marks)
7. A readme.txt file explaining features completed, expected mark and locating presence of codes from other sources. (10 marks)

Some suggestions

1. Regularly spend time for logic-building, design, codes and overall work.
2. Reflect your regularly explored tasks into your assignment.
3. There is quite a lot of works needed for the assignment. Please do not think you can leave the task until the last week before the deadline.