CIT 5940 - Module 4 Extra Credit
Java Streams

Contents

Assignment Overview

Now that you have finished M4PA, you can actually try to simplify your code using lambdas and the Stream API.

Background

If you're not familiar with lambdas or streams (sometimes called sequences), this challenge will have a huge learning curve, but it will also introduce you to a completely different model of programming that is sometimes used in modern Java, and very often used in programming languages like Scala (which also compiles to the JVM), JavaScript, and others.

The high-level goal of this challenge is to re-write your code into a style that uses no loops or other imperative constructs by making each of the six required functions into a pipeline of aggregate operations over a stream pipeline.

1 Setup

Download the starter code file from Canvas. This contains the unimplemented methods for the code that you will write in this assignment.

2 Requirements

2.1 Structure and Compiling

1. You MUST use a JDK version of 17 or higher.
2. You MAY use a JDK version higher than 17, but you MUST set the Java language level to 17 for compatibility with Codio.
3. You MUST NOT change the signatures of any methods.
4. You SHOULD create JUnit tests to help validate your code. These tests SHOULD NOT be a part of your solution. We do expect you to test your code thoroughly with JUnit tests before submission.
5. You MUST fill out the required Academic Integrity signature in the comment block at the top of your submission file.

2.2 Functionality Specifications

2.2.1 General Requirements

1. You MUST NOT use any loops. This includes for, while, do-while, forEach, and any other similar structures.

2. You MUST NOT use any conditional statements. This includes if-else blocks, switch and ternary operator. The only exception to this rule is **`calculateSenteneScore` method** where you MAY use one condition to ensure that result of calculation does not throw exception.
3. You MUST NOT split stream into multiple parts and MUST NOT create temporary variables outside the stream pipeline.
4. You MAY create helper methods and use block lambda expressions as long as they are no more than two lines long.

3 Submission

3.1 Pre-submission Check

Before you submit, please double check that you followed all the instructions, especially the items in the Structure and Compiling section above.

3.2 Codio Submission

1. When you are ready to submit the assignment, go to the Module 4 Programming Assignment Extra Credit Submission item and click the button to go to the Codio platform.
2. Once you are logged into Codio, read the submission instructions in the `README` file. This should be the first thing that appears in the window.
3. Upload your solution and any JUnit test files to the "submit" folder.
4. In the menu bar, select Run JUnit Tests. This will run any unit test files that you have uploaded with your submission. Note that there are no pre-submission checks provided. However you can use this feature as a basic validation check to be sure your code compiles.
5. When you're ready to submit, go to Education and select Mark As Completed. Confirm at the prompt.

4 Grading

4.1 Grading Overview

1. Your submission will be manually graded based on amount of M4PA code built into stream pipeline. This includes null-argument checks, loops, filtering, instantiation of new objects, etc., replaced with or inserted into lambda expressions and stream-related methods.
2. Only methods that pass ALL autograded tests will be further assessed by grading team. All others will be marked as incomplete and will receive 0 points.

3. We will not provide the exact test cases.
4. You have unlimited submissions, until the due date of the assignment as noted in Canvas (or your approved extension).

4.2 Rubric Items

This assignment will be marked on 5 rubric items.

- **`readFile` method** : 4 points
- **`allOccurrences` method** : 2 points
- **`wordTallies` method** : 6 points
- **`calculateScores` method** : 2 points
- **`calculateSenteneScore` method** : 5 points

Total : 19 points

5 Additional Resources

Start here with learning about lambda expressions:
`https://dev.java/learn/lambda-expressions/`
Then move on to learning about the Stream API:
`https://dev.java/learn/the-stream-api/`
The instructor will give a tutorial on lambda expressions and Stream APIs during the Open Office Hour. You can watch the recordings if you cannot attend it.

Good Luck!