Autoform DM Java Developer Test

2021-06-16

Formpipe.

1 Introduction.

This document outlines a task to be completed as a test of your knowledge of Java and general programming skills. When completing this task it is important to demonstrate good programming practices such as:

- Readability
- Reuse
- Structure
- Testing

When submitting your solution, include all the files required to compile the project and anything you have used to test your solution.

The submission will be assess and you will be expected to describe your rationale during the full interview.

1.1 CharacterReader

An interface has been defined called `CharacterReader`. It defines a single method which returns a character from a stream. An `EndOfStreamException` is thrown if there are no more characters to read.

There are two implementations of `CharacterReader` given, one for each part of the following task.

1.2 Guidelines

You must complete the tasks using the given code.

You may add as many classes/methods as you need, but do not modify the signatures of CharacterReader or its implementations.

You may implement your solution in Java, Scala or any other compatible language.

Feel free to use Maven for any dependencies you require.

2 Task

2.1 Question 1

Write an application that takes a `CharacterReader` and outputs a list of word frequencies ordered first by word count and then alphabetically.

You should use the **Simple**`CharacterReader` as the implementation class for this.

For example if the stream returned "It was the best of times, it was the worst of times" then the output would be:

```
It – 2
Of – 2
The – 2
Times – 2
Was – 2
Best – 1
Worst – 1
```

2.2 Question 2

Write an application that takes an array of `CharacterReader`'s, accesses them in parallel, and outputs the combined word counts (as in the first question), every 10 seconds until all threads are complete.

You should use the **Slow**`CharacterReader` as the implementation class for this.

For example if the stream returned "It was the best of times, it was the worst of times" and your array contained 10 SlowCharacterReaders, after the first 10 seconds you might print:

```
It – 10
Of – 10
The – 10
Times – 10
Was – 10
Best – 10
```

And after another 10 seconds would print:

```
It – 20
Of – 20
The – 20
Times – 20
Was – 20
Best – 10
Worst - 10
```

Then complete.