Lab 6

1. **Accept** the **Lab 6** assignment using GitHub Classroom link with your Github account.

2. Clone your GitHub Classroom repository using IntelliJ IDEA.

3. Complete **Lab6 exercises 1-3** as described in the comments and the requirements in this file.

4. Test your solutions in the **main functions**.

5. Once your code runs correctly in **main**, use the provided **JUnit tests** to verify your work.

- If the tests fail, use the **debugger** to fix your code.
- **Do not modify the JUnit test files. Any changes will be detected.**

6. Use **Git** to submit your code to your GitHub Classroom repository.

- Check the **GitHub Actions results** to confirm that your code passes all tests.

7. After completing all the steps above, be prepared to **demo** your code to me.

Overview

In this lab, you will use File IO, Java Collection class, HashMap and read and write from text files.

Exercise 1

In this exercise, you will use the program in DataManager.java. The **DataManager** class in this program has methods to read sales data from a text file, parse the data and store it in a HashMap. A HashMap in Java works on hashing principle. It is a data structure which allows us to store an object and retrieve it in constant time O(1) provided we know the key. For more information on HashMap's.

Open the text file: ***sales.txt*** and examine the data. Each line of text in the text file has 3, colon-separated fields, namely: "year", "quarter" and "sales" in thousands. The method, *readDataFromFile(),* reads each line, splits the line into 3 fields, makes a key using the year and quarter and stores the sales number (third field) in the HashMap at that key. Complete the program as per comments.

a) Complete the constructor for "Sales(**int** quarter, **int** year, **int** salesInK)"

b) Complete the *makeKey()* method using the given comments.

c) Complete the *readDataFromFile()* method using the given comments.

d) The code in the *main()* should work correctly after you compile and run the program.
Here, you are retrieving the Sales object stored at a specific key.

Exercise 2

For the class **FileProcessor** implements the static method *fileCopy().*

a) Read the file: **input.txt**.

b) Finish the implementation of *fileCopy()*.

c) Compare the files, the number of bytes should be the same, if the *fileCopy()* method worked.

Exercise 3

In this exercise, you will complete the static method *wordCount()* in the class, **FileProcessor**. This method should open the **input.txt** file, read the contents line by line and split each line into words (tokens) using space as the delimiter. It should write each line number and the number of words on that line into the output file. Remove the comments around this method, complete the method and test it by calling it from main.

The format of the output in the output file should be similar to the one below (of course, the no. of words in each line depends on your actual input in the input file).

Input File: input.txt

Line 0 No. of words: 16
Line 1 No. of words: 18
Line 2 No. of words: 17
Line 3 No. of words: 16
Line 4 No. of words: 17
Line 5 No. of words: 17
Line 6 No. of words: 17