Assignment-4 [PREVIEW-MODE]

Assignment 4

Due date

- 11.59 PM EST, July 20th

Git url

- https://classroom.github.com/a/5ctzf7o1

Submit your code as per the provided instructions.

Assignment Goal

Apply the design principles you have learned so far to develop software for the given problem.

Team Work

- ~~You are required to work in a team of 2 students for this project.~~ You are required to work ALONE on this project

Programming Language

You are required to use Java.

Compiling and Running Commands

- Compilation: Your code should compile on remote.cs.binghamton.edu
- You are required to use ANT as the build tool for this assignment.

Policy on sharing of code

EVERY line of code that you submit in this assignment should be written by members of your team. Do NOT show your code to any other group. Our code-comparison software can very easily detect similarities.

P[illegible]o piazza if you have any questions about the requirements. Do not send emails. Do not post your code asking for help with debugging. However, it is okay to post design/concept questions on programming in and also in the documentation of every source file in which that code appears.

Project Description

Program with visitors to determine features in two input files that have integers.

- The input for the program consists of two separate input files, both with a positive 2 digit integer in each line. Assume that the input may contain duplicates in the individual files.
- Create an interface MyArrayI that defines an API for an Abstract Data Type (ADT) representing an array. In addition, MyArrayI is an *Element*.
- Define an ADT, MyArray, that implements MyArrayI and stores an internal array of integers. The internal array should be created with an initial capacity of 10 and increased by 50% in capacity each time an integer has to be added beyond the current capacity.
- Each MyArray object will store the integers from a single input file.
- Design a visitor, PopulateMyArrayVisitor, that reads from a file and populates MyArray one integer at a time. The visitor should use an instance of the FileProcessor to read from the input file one line at a time (one integer at a time). Check for the boundary condition that the input file does not exist or is empty. Throw an exception and exit if a string from the file cannot be converted to an integer. Other than that, assume the input file is well formed and no other exceptions need to be handled.
- The PopulateMyArrayVisitor can take the name of the input file in its constructor or have a setX(..) method for it. You need to apply this visitor once for each of the two input files to get two instances of MyArray that are populated with integers.
- Create an interface MyArrayListI that defines an API for a ADT representing an arraylist. In addition, MyArrayListI is an *Element*.
- Define an ADT, MyArrayList, that implements MyArrayListI and maintains an internal array of the MyArray objects.
- Design a visitor, CommonIntsVisitor, that determines the integers that are common in the two ADTs stored in MyArrayList, and stores those integers (without duplicates) in an appropriate data structure in Results.
- Design a visitor, MissingIntsVisitor that determines the 2 digit integers (between 00 and 99) that are missing in MyArray and stores them in an appropriate data structure in Results.
- The output files should contain a single integer per line.
- Use a singleton Logger and design your own debugging scheme.
- The driver should accept the input file names, output file names, and debug value, via the command line.
    1. *-Dinput1* - First input file containing 2 digits integers per line.
    2. *-Dinput2* - Second input file containing 2 digits integers per line.
    3. *-Dcommonintsout* - Output file to store results of applying CommonIntsVisitor.
    4. *-Dmissingintsout* - Output file to store results of applying MissingIntsVisitor.
    5. *-Ddebug* - Debug value.
- The driver should do the following:
    - Create required instances of Results.
    - Create instance of FileProcessor.
    - Create instances of the visitors.
    - Create two instances of MyArray.
    - Use the PopulateMyArrayVisitor to populate two instances of MyArray.
    - Apply CommonIntsVisitor to determine common ints in ADTs stored in MyArrayList.
    - Apply MissingIntsVisitor separately to each of the MyArray instances.
    - Call appropriate methods in Result instances to print the output of each of the visitors.

ⓘ Helpers:

1. There is only a single visitor interface.
2. Unlike the traditional viistor pattern where there is just a single visit(..) method in the Visitor

defines the API for the ADT.

4. No business logic should be written in the driver code.
5. **ADT design**
   - Empty constructor.
   - Explicit value constructor.
   - Getters and Setters.
   - Override toString.
   - Empty finalize method.
   - Override Clone.
6. **Driver code**
   - Driver code should not include business logic.
   - Simple and concise.

NOTES ON GRADING

- Class participation points will be given to students who post good questions on piazza seeking clarification on the assignment or finding ambiguities, and also to those who respond and participate to help make the discussion interesting and informative.
- Class participation points will be given to students who also post sample interesting input examples with corresponding output and metrics.

Sample Input Files sent by students in this course

Please check piazza.

Compiling and Running Java code

- Your submission must include a readme in markdown format with the name **README.md**.
- Your README.md file should have the following information:
  - instructions on how to compile the code
  - instructions on how to run the code
  - justification for the choice of data structures (in terms of time and/or space complexity).
  - citations for external material utilized.

Code Organization

- You should have the following directory structure (replace username with github username).
- ./csx42-summer-2020-assign4-username
- ./csx42-summer-2020-assign4-username/README.md
- ./csx42-summer-2020-assign4-username/.gitignore
- ./csx42-summer-2020-assign4-username/arrayvisitors
- ./csx42-summer-2020-assign4-username/arrayvisitors/src
- ./csx42-summer-2020-assign4-username/arrayvisitors/src/build.xml
- ./csx42-summer-2020-assign4-username/arrayvisitors/src/arrayvisitors
- ./csx42-summer-2020-assign4-username/arrayvisitors/src/arrayvisitors/driver
- ./csx42-summer-2020-assign4-username/arrayvisitors/src/arrayvisitors/driver/Driver.java
- ./csx42-summer-2020-assign4-username/arrayvisitors/src/arrayvisitors/adt
- ./csx42-summer-2020-assign4-username/arrayvisitors/src/arrayvisitors/adt/MyArrayI.java
- ./csx42-summer-2020-assign4-username/arrayvisitors/src/arrayvisitors/adt/MyArrayListI.java
- ./csx42-summer-2020-assign4-username/arrayvisitors/src/arrayvisitors/adt/MyArray.java
- ./csx42-summer-2020-assign4-username/arrayvisitors/src/arrayvisitors/adt/MyArrayList.java
- ./csx42-summer-2020-assign4-username/arrayvisitors/src/arrayvisitors/visitors