A01 - Arrays

9/9/2023

MISSING 0/30 Points

Attempt 1 | Review Feedback | Offline Score: **0/30** | Add Comment

Unlimited Attempts Allowed
8/26/2023 to 9/16/2023

Details

Fundamentals

Assignment: A01

Learning Objectives

- Build algorithms using one- and two-dimensional arrays
- Review programming concepts practiced in CSIS 1400 and CSIS 1410.

Overview

A01 is designed to review and build on skills you developed in CSIS-1400 and CSIS-1410 by building algorithms that solve challenges using one- and two-dimensional arrays.

When grading assignments, I primarily look at the correctness of the code. However, I also consider code quality, especially whether the posted style guidelines and best practices are followed. You can find them under **Guidelines and Expectations** (https://slcc.instructure.com/courses/915963/pages/guidelines-and-expectations) in the Orientation Module. Please review and follow them on all programming assignments in this course.

Also, consider reading the rubric before starting to work on assignments. It helps you focus your efforts and maximize your potential for success.

Instruction

1. **Setting Up Your Project:**
   Create a new Java project called **A01_Arrays** with a package named **a01**.
   Download the files **A01Version1.java** (https://slcc.instructure.com/courses/915963/files/149887334?wrap=1)

(https://slcc.instructure.com/courses/915963/files/149887334/download?download_frd=1) and A01Version2.java (https://slcc.instructure.com/courses/915963/files/149887332?wrap=1) (https://slcc.instructure.com/courses/915963/files/149887332/download?download_frd=1) and add them to the package a01.
Replace the ..... in the doc comment with your name.

2. **Choose the class**
   You need to implement all methods of one class, but you choose which one you want to work on.
   - Each class includes four methods: two use one-dimensional arrays, and two use two-dimensional arrays.
   - There are also differences:
     A01Version1 includes foundational challenges that provide you with the review you need. A01Version2 goes a step further for those who like to challenge themselves.

3. **Dynamic Object Viewer**
   Install the jGrasp Plugin for Eclipse (https://slcc.instructure.com/courses/915963/pages/jgrasp-plugin-for-eclipse) to access jGrasp's dynamic object viewer. It can safe you a lot of time when troubleshooting the code.
   FYI: jGrasp also has a plugin for IntelliJ.

4. **Implement the methods**
   - Read all the information in the doc comment including the @throws tags that describe how to handle invalid input.
     Each text line of the doc comment starts with an asterisk and a space. If that is not what you see, adjust the formatting so you can easily read the specifications and examples.
   - Replace the lines with the // TODO comment with a method implementation that matches the specifications in the doc comments.
   - Optional help:
     This introduction to different types of matrices tells you what square matrices (https://www.mathsisfun.com/algebra/matrix-types.html#:~:text=A%20square%20matrix%20has%20the%20same%20number%20of%20rows%20as%20columns.) are.

5. **Write test code in the main method**
   - For each of the four methods you implemented, test every example listed in the doc comments and all special cases (how to handle an empty array, situations requiring exceptions, etc.)
   - Include labels clearly showing the method you are testing and the input and output values.

6. **JUnit tests**
   This assignment includes JUnit tests to help you provide evidence that your code works as expected.
   Run these JUnit tests after using your own test code to convince yourself that your code meets the requirements. This way, you get feedback about the correctness of your implementation and also about the quality of your testing.
   If you encounter a JUnit test that doesn't pass, troubleshoot your code. Additionally, look at your test code and identify what you can change so similar bugs will be caught next time.
   Here are two test files:
   A01_Version1Test.java (https://slcc.instructure.com/courses/915963/files/149887331?wrap=1) (https://slcc.instructure.com/courses/915963/files/149887331/download?download_frd=1) and A01_Version2Test.java (https://slcc.instructure.com/courses/915963/files/149887333?wrap=1) (https://slcc.instructure.com/courses/915963/files/149887333/download?download_frd=1)
   Please download the one corresponding to the class you chose to implement and add it to the package a01.

Submission

(https://community.canvaslms.com/t5/Student-Guide/How-do-I-embed-images-from-Canvas-into-the-Rich-Content-Editor/ta-p/356)

- **Attach the file(s) that include your code**
- **Create and embed a Video: (1.5 - 3 min)**
  Follow the **Guidelines for Assignment/CE Recordings** (https://slcc.instructure.com/courses/915963/pages/guidelines-and-expectations) and include the following:
  - A title page and a brief introduction
  - Show the code
    That should include the methods you implemented, your test code, and the provided JUnit tests (unmodified).
  - Run your test code and show the results.
  - Run the unmodified JUnit tests that were provided and clearly display the test results.

(https://community.canvaslms.com/t5/Student-Guide/How-do-I-embed-images-from-Canvas-into-the-Rich-Content-Editor/ta-p/356)

(https://community.canvaslms.com/t5/Student-Guide/How-do-I-embed-images-from-Canvas-into-the-Rich-Content-Editor/ta-p/356)

View Rubric

2420 - A01					
Criteria	**Ratings**				**Pts**
first method (1d array) view longer description	**5 pts** **Well done!**	**4 pts** **Close** All JUnit tests passed except for one.	**0 pts** **No Marks**		/ 5 pts
second method (1d array) view longer description	**5 pts** **Well done!**	**4 pts** **Close** All JUnit tests passed except for one.	**0 pts** **No Marks**		/ 5 pts
third method (2d array) view longer description	**5 pts** **Well done!**	**4 pts** **Close** All JUnit tests passed except for one.	**0 pts** **No Marks**		/ 5 pts
fourth method (2d array) view longer description	**5 pts** **Well done!**	**4 pts** **Close** All JUnit tests passed except for one.	**0 pts** **No Marks**		/ 5 pts
Test Code view longer description	**5 pts** **Well done!**	**4 pts** **Close**	**2 pts** **Getting There**	**0 pts** **No Marks**	/ 5 pts
Style | Best Practices | Video view longer description	**5 pts** **Well done.**	**4 pts** **Close**	**0 pts** **No Marks**		/ 5 pts
					Total Points: 0