Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-002-S2024/it114-m2-java-problems/grade/mth39

IT114-002-S2024 - [IT114] M2 Java Problems

Submissions:

Submission Selection

1 Submission [active] 2/3/2024 8:19:53 PM

Instructions

COLLAPSE

Guide:

1 .Make sure you're in the main branch locally and `git pull origin main` any pending changes
2 .Make a new branch per the recommended branch name below (git checkout -b ...)
3 .Grab the template code
  from https://gist.github.com/MattToegel/fdd2b37fa79a06ace9dd259ac82728b6
4 .Create individual Java files for each problem and save the files inside a subfolder of your choice
    1 .The should end with the file extension in lowercase .java
5 .Move the unedited template files to github
    1 .`git add .`
    2 .`git commit -m "adding template files`
    3 .`git push origin <homework branch>` (see below and don't include the < >)
    4 .Create and open a pull request from the homework branch to main (leave it open until later steps)
6 .Note: As you work, it's recommended to add/commit at least after each solution is done (i.e., 3+ times in this case)
    1 .Make sure the files are saved before doing this
7 .Fill in the items in the worksheet below (save as often as necessary)
8 .Once finished, export the worksheet
9 .Add the output file to any location of your choice in your repository folder (i.e., a Module2 folder)
10Check that git sees it via `git status`
11If everything is good, continue to submit
    1 .Track the file(s) via `git add`
    2 .Commit the changes via `git commit` (don't forget the commit message)
    3 .Push the changes to GitHub via `git push` (don't forget to refer to the proper branch)
    4 .Create a pull request from the homework related branch to main (i.e., main <- "homework branch")
    5 .Open and complete the merge of the pull request (it should turn purple)
    6 .Locally checkout main and pull the latest changes (to prepare for future work)
12Take the same output file and upload it to Canvas
    1 .*This step is new since GitHub renders the PDF as an image the links aren't clickable so this method works better
    2 .*Remember, the github process of these files are encouragement for your tracking of your progress

Branch name: M2-Java-Problems

Tasks: 8 Points: 10.00

COLLAPSE

Problem 1 (3 pts.)

COLLAPSE

Task #1 - Points: 1

Text: Screenshot of the Problem 1 Solved Code and Output

Details:
Only make edits where the template code mentions.

Solution should ensure that any passed in array will have only the odd values output.
Requires at least 2 screenshots (code + output from terminal)

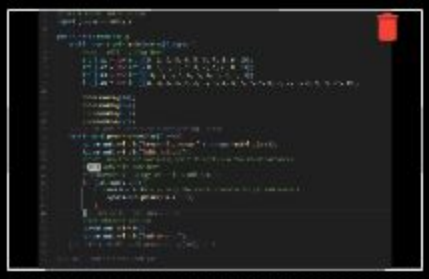
Checklist

*The checkboxes are for your own tracking

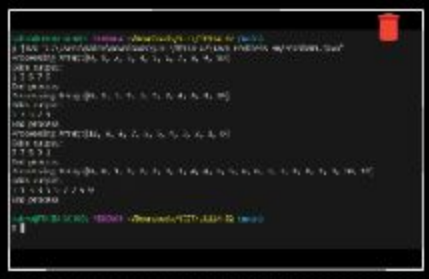
#	Points	Details
☐ #1	1	Edits were done only in the processArray() method and original template code/comments remain untouched
☐ #2	1	Only arr is used (no direct usage of a1, a2, a3, a4)
☐ #3	5	Only odd values output (not odd indexes/keys)
☐ #4	1	Includes code comments with student's ucid and date
☐ #5	1	Terminal output is fully visible

Task Screenshots:

Large Gallery



Checklist Items (0)



Checklist Items (0)

Showing screenshot of completion of problem 1

Showing screenshot of terminal output of problem 1

COLLAPSE

Task #2 - Points: 1

Text: Explain your solution

Checklist *The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	Clearly explains how the code/logic solves the problem (mentions how the odd values are determined)

Response:

Within the for loop, the if statement checks whether the current number (num) is odd or not by using 'num % 2 != 0'. This condition uses the modulo operator (%) to calculate the remainder when num is divided by 2. If the remainder is not zero, it means that the number is odd. If the condition is true (the number is odd), the code executes the block inside the if statement. Inside the if block, System.out.print(num + " "); prints the current odd number followed by a space, without moving to the next line.

● COLLAPSE

Problem 2 (3 pts.)

● COLLAPSE

Task #1 - Points: 1

Text: Screenshot of the Problem 2 Solved Code and Output

ⓘ Details:

Only make edits where the template code mentions.

Solution should ensure that any passed in array will have its values converted to a positive version of the value AND converted back to the original data type.
Requires at least 2 screenshots (code + output from terminal)

Checklist *The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	Edits were done only in the getTotal() method and original template code/comments remain untouched (unless noted)
☐ #2	1	Only arr is used (no direct usage of a1, a2, a3, a4)
☐ #3	5	Passed in array's values get summed AND rounded to two decimal places like currency (i.e., 0.00, 0.10, 1.10)
☐ #4	1	Includes code comments with student's ucid and date
☐ #5	1	Terminal output is fully visible

Task Screenshots:

Large Gallery

Checklist Items (0)

Checklist Items (0)

Screenshot showing completion of problem 2

Screenshot showing terminal output of problem 2

COLLAPSE

Task #2 - Points: 1

Text: Explain your solution

Checklist

*The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	Clearly explains how the code/logic solves the problem (mentions both how the values get summed and how the rounding is solved correctly)

Response:

The for loop iterates through each double value in the input array and each value is added to the total variable. 'Math.round(total * 100.0) / 100.0' multiplies the total by 100 to shift the decimal point two places to the right, rounds the result to the nearest whole number, and then shifts the decimal point back to its original position, rounding the total to two decimal places. The rounded total value is then assigned to the totalOutput variable as a string (totalOutput = total+"").

COLLAPSE

Problem 3 (3 pts.)

COLLAPSE

Task #1 - Points: 1

Text: Screenshot of the Problem 2 Solved Code and Output

Details:

Only make edits where the template code mentions.

Solution should ensure that any passed in array will have its values converted to a positive version of the value AND converted back to the original data type.
Requires at least 2 screenshots (code + output from terminal)

Checklist

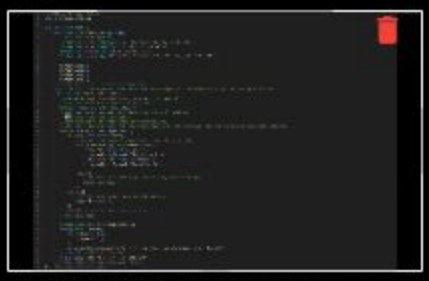
*The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	Edits were done only in the bePositive() method and original template code/comments remain untouched
☐ #2	1	Only arr is used (no direct usage of a1, a2, a3, a4)
☐ #3	1	Passed in array's values will get converted to a positive version AND converted back to the original data

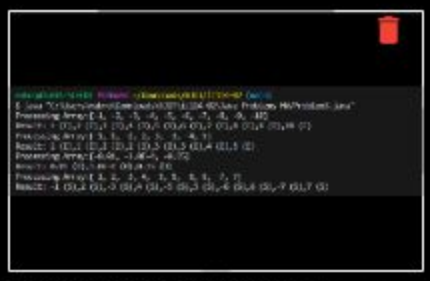
☐ #3	5	Passed in array's values will get converted to a positive version AND converted back to the original data type
☐ #4	1	Includes code comments with student's ucid and date
☐ #5	1	Terminal output is fully visible

Task Screenshots:

Large Gallery



Checklist Items (0)



Checklist Items (0)

Screenshot showing the completion of problem 3

Screenshot showing the terminal output of problem 3

COLLAPSE

Task #2 - Points: 1

Text: Explain your solution

Checklist

*The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	Clearly explains how the code/logic solves the problem (mentions both the conversion to positive and conversion to original data type)

Response:

The for loop iterates through each element of the input array. Inside the loop, it checks if the current element is an instance of the Number class, ensuring that it's a numeric value. If the value is negative, it proceeds to handle the conversion to a positive value. Depending on the original data type of the numeric value (checked with instanceof), it uses Math.abs to convert it to its positive equivalent. If it's an integer, or double it converts it to a positive value. If the value is already positive or zero, it retains the original value. If the current element is not a numeric value (fails the instanceof Number check), it keeps the original value unchanged.

COLLAPSE

Reflection (1 pt.)

COLLAPSE

Task #1 - Points: 1

Text: Reflect on your experience

Details:

Talk about any issues you had, how you resolved them, and anything you learned during this process.

Provide concrete details/examples.

Response:

I gained a better understanding of how to use loops in Java.

COLLAPSE

Task #2 - Points: 1

Text: Include the pull request link for this branch

Details:

The correct link will end with /pull/ and a number.