Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-450-M2024/it114-module-2-java-problems/grade/yh68

IT114-450-M2024 - [IT114] Module 2 Java Problems

Submissions:

Submission Selection

1 Submission [active] 6/1/2024 1:23:38 AM

Instructions

COLLAPSE

Overview Video: https://youtu.be/4M8Di5jrcZQ

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. They 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 branch_name` (see below).
    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).
10. Check that git sees it via `git status`.
11. If 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).

12. Take the same output file and upload it to Canvas.

Branch name: M2-Java-Problems

Tasks: 6 Points: 10.00

Problem 1 (3 pts.)

^COLLAPSE^

^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)

#1) Screenshot the output of the solved problem

```
Yahya@YousefsLaptop MINGW64 ~/IT114 Projects/yh68-it114-450/M2 (M2-JavaProbs)
$ java Problem1.java
Processing Array:[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Odds output:
Printing my odd values...
1 3 5 7 9
End process
Processing Array:[0, 1, 3, 5, 7, 9, 2, 4, 6, 8, 10]
Odds output:
Printing my odd values...
1 3 5 7 9
End process
Processing Array:[10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
Odds output:
Printing my odd values...
9 7 5 3 1
End process
Processing Array:[0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10]
Odds output:
Printing my odd values...
1 1 3 3 5 5 7 7 9 9
End process
```

Caption (required) ✓

Describe/highlight what's being shown

Showing the correct output for Problem1.java

#2) Screenshot the code solution (ucid/date must be included as a comment)

```
static void processArray(int[] arr){
    System.out.println("Processing Array:" + Arrays.toString(arr));
    System.out.println(x:"Odds output:");
    //hint: use the arr variable; don't diretly use the a1-a4 variables
    // UCID & Date: yh68 6/1/2024
    //TODO add/edit code here
    System.out.println(x:"Printing my odd values...");
    for (int num : arr) {
        if (num % 2 != 0) {
            System.out.print(num + " ");
        }
    }
    //end add/edit section
    System.out.println();
    System.out.println(x:"End process");
}       You, 44 minutes ago • adding template files
```

Caption (required) ✓

Describe/highlight what's being shown

Showing that I solved this using a for each loop

Explanation (required) ✓

Explain in concise steps how this logically works

EDIT RESPONSE

It works by iterating over each element in the array. The if statement checks if the number is odd or even. Then it prints.

Problem 2 (3 pts.)

COLLAPSE

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 summed AND the final result converted to two decimal places (i.e., 0.10, 1.00, 1.01).
Requires at least 2 screenshots (code + output from terminal)

#1) Screenshot the output of the solved problem

```
Yahya@YousefsLaptop MINGW64 ~/IT114 Projects/yh68-it114-450/M2 (M2-JavaProbs)
$ javac Problem2.java

Yahya@YousefsLaptop MINGW64 ~/IT114 Projects/yh68-it114-450/M2 (M2-JavaProbs)
$ java Problem2.java
Processing Array:[10.001, 11.591, 0.011, 5.991, 16.121, 0.131, 100.981, 1.001]
Adding values to total variable
Displaying output as two decimal places:
Total is 145.83
End process
Processing Array:[1.99, 1.99, 0.99, 1.99, 0.99, 1.99, 0.99, 0.99]
Adding values to total variable
Displaying output as two decimal places:
Total is 11.92
End process
Processing Array:[0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01]
Adding values to total variable
Displaying output as two decimal places:
Total is 0.10
End process
Processing Array:[10.01, -12.22, 0.23, 19.2, -5.13, 3.12]
Adding values to total variable
Displaying output as two decimal places:
Total is 15.21
End process
```

Caption (required) ✓

Describe/highlight what's being shown

Showing the correct output for Problem2.java

#2) Screenshot the code solution (ucid/date must be included as a comment)

```
// UCID & Date: yh68 6/1/2024
//TODO add/edit code here
System.out.println(x:"Adding values to total variable");
for (double num : arr) {
    total += num;
}       You, 18 minutes ago • example solved each problem added .git
//set the double to a string variable
//TODO ensure rounding is to two decimal places (i.e., 0.10, 0.01,
System.out.println(x:"Displaying output as two decimal places:");
totalOutput = String.format(format:"%.2f", total);
//end add/edit section
```

Caption (required) ✓

Describe/highlight what's being shown

Showing that I solved this using a for each loop and the String.format() method

Explanation (required) ✓

Explain in concise steps how this logically works

PREVIEW RESPONSE

The for each loop iterates over each element in the array. The second line just adds the element to the total. The last line uses the String.format() method to format the double total to two decimal places, the %f is a format specifier for floating point numbers so %.2f means two decimal places.

Problem 3 (3 pts.)

^COLLAPSE^

^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)

#1) Screenshot the output of the solved problem

```
Yahya@YousefsLaptop MINGW64 ~/IT114 Projects/yh68-it114-450/M2 (M2-JavaProbs)
$ javac Problem3.java

Yahya@YousefsLaptop MINGW64 ~/IT114 Projects/yh68-it114-450/M2 (M2-JavaProbs)
$ java Problem3.java
Processing Array:[-1, -2, -3, -4, -5, -6, -7, -8, -9, -10]
Making each value positive
Assigning each value to the output array in the same index as the original data type
Result: 1 (I),2 (I),3 (I),4 (I),5 (I),6 (I),7 (I),8 (I),9 (I),10 (I)
Processing Array:[-1, 1, -2, 2, 3, -3, -4, 5]
Making each value positive
Assigning each value to the output array in the same index as the original data type
Result: 1 (I),1 (I),2 (I),2 (I),3 (I),3 (I),4 (I),5 (I)
Processing Array:[-0.01, -1.0E-4, -0.15]
Making each value positive
Assigning each value to the output array in the same index as the original data type
Result: 0.01 (D),1.0E-4 (D),0.15 (D)
Processing Array:[-1, 2, -3, 4, -5, 5, -6, 6, -7, 7]
Making each value positive
Assigning each value to the output array in the same index as the original data type
Result: 1 (S),2 (S),3 (S),4 (S),5 (S),5 (S),6 (S),6 (S),7 (S),7 (S)
```

Caption (required) ✓

Describe/highlight what's being shown

Showing the correct output for Problem3.java

#2) Screenshot the code solution (ucid/date must be included as a comment)

```
// UCID & Date: yh68 6/1/2024
System.out.println(x:"Making each value positive");
for (int i = 0; i < arr.length; i++) {
    if (arr[i] instanceof Integer) {
        output[i] = Math.abs((Integer) arr[i]);
    } else if (arr[i] instanceof Double) {
        output[i] = Math.abs((Double) arr[i]);
    } else if (arr[i] instanceof String) {
        int intValue = Integer.parseInt((String) arr[i]);
```

```
            output[i] = String.valueOf(Math.abs(intValue));
        } else {
            output[i] = arr[i];
        }
    }

    System.out.println(x:"Assigning each value to the output array in t
    //end edit section
```

Caption (required) ✓

Describe/highlight what's being shown

Showing that I used a for loop, as well as if/else if/else statements to solve this problem

Explanation (required) ✓

Explain in concise steps how this logically works

PREVIEW RESPONSE

The for loop iterates until it reaches the length of the array. Inside the loop each element of the array is checked to determine its type using instanceof if the element is an integer Math.abs() is used to get its absolute value else if the element is a double Math.abs() is used to get its absolute value else if the element is a string it gets parsed to an integer using Integer.parseInt() then once it gets the absolute value it is converted back to a string using String.valueOf() else the element's type is not integer double or string so it just uses the original value in the output

Reflection (1 pt.)

COLLAPSE

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. At least a few sentences.

Response:

I didn't learn much regarding gitbash in this lesson. I did learn a little about < T > in java, although I don't know too much about it yet. No issues thank God.

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.

URL #1
https://github.com/FreePalestine7/yh68-it114-450/pull/8

COLLAPSE

Task #3 - Points: 1

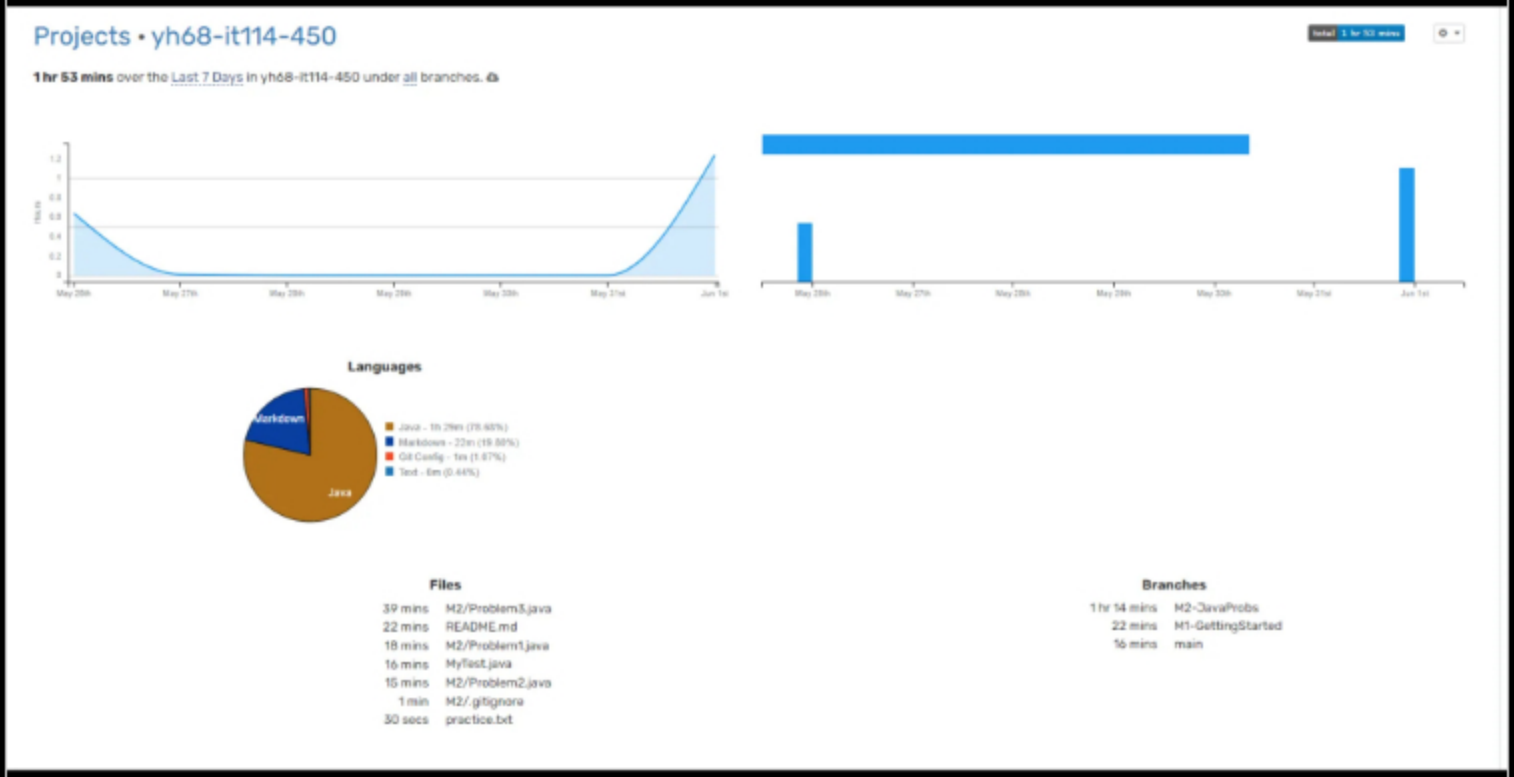
Text: Add Screenshot of Wakatime

Details:
Note: The duration of time isn't directly related to the grade, the goal is to just make sure time is being tracked

Task Screenshots:

Gallery Style: Large View

Small Medium Large



Showing overall and specific times. I heard you say in the video that you were more focused on the bottom and overall time so I used webdev tools and took the stuff we didn't need to screenshot out so I don't have to take two screenshots lol.

End of Assignment