Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-451-M2024/it114-module-2-java-problems/grade/st278

IT114-451-M2024 - [IT114] Module 2 Java Problems

Submissions:

Submission Selection

1 Submission [active] 6/7/2024 1:43:47 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

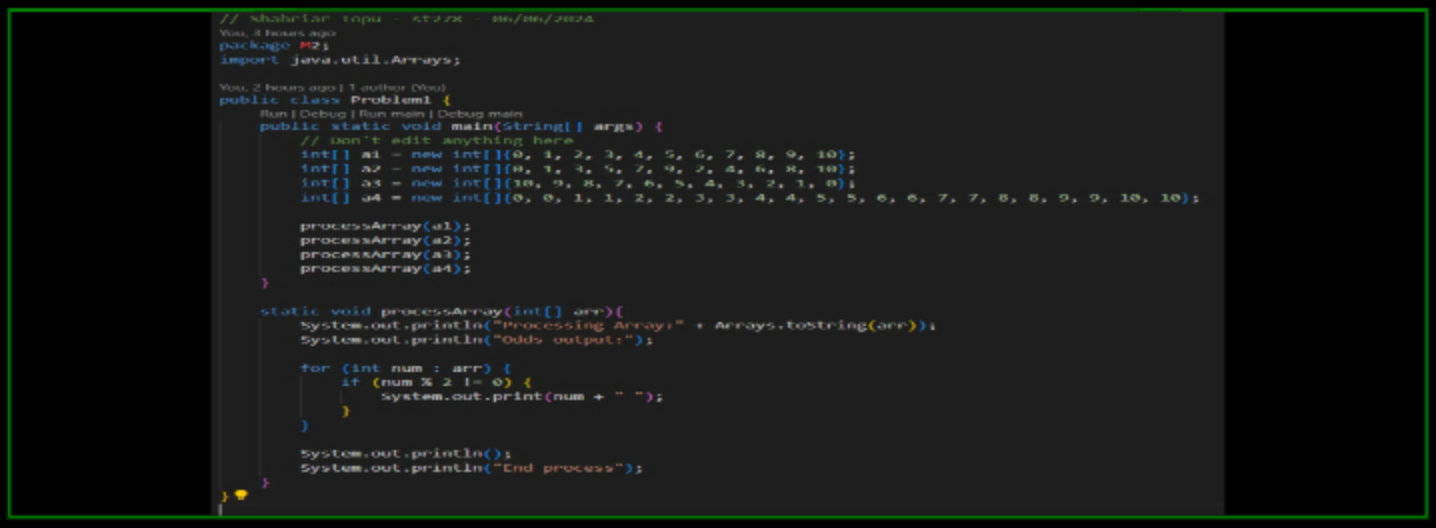
```
Shahriar Topu@LapTopu MINGW64 /c/JuniorYear/IT114/st278-IT114-M2024 (M2-Java-Problems)
$ java M2.Problem1
Processing Array:[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Odds output:
1 3 5 7 9
End process
Processing Array:[0, 1, 3, 5, 7, 9, 2, 4, 6, 8, 10]
Odds output:
Processing Array:[0, 1, 3, 5, 7, 9, 2, 4, 6, 8, 10]
Odds output:
1 3 5 7 9
End process
Processing Array:[10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
Odds output:
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:
1 1 3 3 5 5 7 7 9 9
End process
```

Caption (required) ✓

Describe/highlight what's being shown

The output of the first problem.

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



```
// Shahriar Iqbal - sti2k - mm/dd/2024
You, 4 hours ago
package M2;
import java.util.Arrays;

You, 2 hours ago | 1 author (You)
public class Problem1 {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        // don't edit anything here
        int[] a1 = new int[]{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
        int[] a2 = new int[]{0, 1, 4, 5, 7, 9, 2, 4, 6, 8, 10};
        int[] a3 = new int[]{10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0};
        int[] a4 = new int[]{0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10};

        processArray(a1);
        processArray(a2);
        processArray(a3);
        processArray(a4);
    }

    static void processArray(int[] arr){
        System.out.println("Processing Array:" + Arrays.toString(arr));
        System.out.println("Odds output:");

        for (int num : arr) {
            if (num % 2 != 0) {
                System.out.print(num + " ");
            }
        }

        System.out.println();
        System.out.println("End process");
    }
}
```

Caption (required) ✓

Describe/highlight what's being shown

The code with the necessary information for problem 1.

Explanation (required) ✓

Explain in concise steps how this logically works

EDIT RESPONSE

The processArray method takes an integer array, prints the array, then prints only the odd numbers from the array. Finally, it prints an end-of-process message.

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

```
Shahriar Topu@LapTopu MINGW64 /c/JuniorYear/IT114/st278-IT114-M2024 (M2-Java-Problems)
$ java M2.Problem2
Processing Array:[10.001, 11.591, 0.011, 5.991, 16.121, 0.131, 100.981, 1.001]
Total is 145.83
End process
Processing Array:[1.99, 1.99, 0.99, 1.99, 0.99, 1.99, 0.99, 0.99]
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]
Total is 0.10
End process
Processing Array:[10.01, -12.22, 0.23, 19.2, -5.13, 3.12]
Total is 15.21
End process
```

Caption (required) ✓

Describe/highlight what's being shown

The output of the second problem.

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

```java
// Shahriar Topu - st278 - 06/06/2024
package M2;
import java.util.Arrays;

public class Problem2 {
    public static void main(String[] args) {
        double[] a1 = new double[]{10.003, 11.591, 0.011, 5.991, 16.121, 0.131, 100.981, 1.001};
        double[] a2 = new double[]{1.99, 1.99, 0.99, 1.99, 0.99, 1.99, 0.99, 0.99};
        double[] a3 = new double[]{0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01};
        double[] a4 = new double[]{10.01, -12.22, 0.23, 19.20, -5.13, 3.12};

        getTotal(a1);
        getTotal(a2);
        getTotal(a3);
        getTotal(a4);
    }

    static void getTotal(double[] arr){
        System.out.println("Processing Array:" + Arrays.toString(arr));
        double total = 0;
            for (double num : arr) {
            total += num;
        }

        String totalOutput = String.format(format:"%.2f", total);

        System.out.println("Total is " + totalOutput);
        System.out.println(x:"End process");
    }
}
```

Caption (required) ✓

Describe/highlight what's being shown

The code with the necessary information for problem 2.

Explanation (required) ✓

Explain in concise steps how this logically works

EDIT RESPONSE

The getTotal method takes a double array, prints the array, calculates the sum of its elements, rounds the total to two decimal places, and prints the rounded total. It then prints an end-of-process like the first questions one does.

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

```
Shahriar Topu@LapTopu MINGW64 /c/JuniorYear/IT114/st278-IT114-M2024 (M2-Java-Problems)
$ java M2.Problem3
Processing Array:[-1, -2, -3, -4, -5, -6, -7, -8, -9, -10]
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]
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]
Result: 0.01 (D),1.0E-4 (D),0.15 (D)
Processing Array:[-1, 2, -3, 4, -5, 5, -6, 6, -7, 7]
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

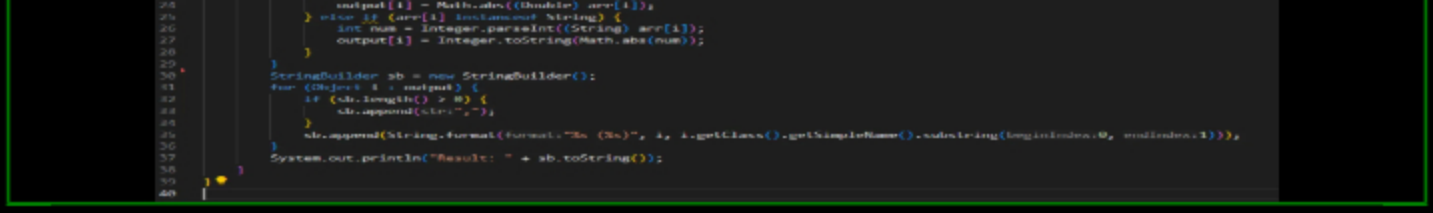
The output of the third problem.

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

```
// Shahriar Topu - st278 - 06/06/2024
You, 3 hours ago
package M2;
import java.util.Arrays;
You, 1 minute ago | 1 author (You)
public class Problem3 {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Integer[] a1 = new Integer[]{-1, -2, -3, -4, -5, -6, -7, -8, -9, -10};
        Integer[] a2 = new Integer[]{-1, 1, -2, 2, 3, -3, -4, 5};
        Double[] a3 = new Double[]{-0.01, -0.0001, -.15};
        String[] a4 = new String[]{"-1", "2", "-3", "4", "-5", "5", "-6", "6", "-7", "7"};

        bePositive(a1);
        bePositive(a2);
        bePositive(a3);
        bePositive(a4);
    }
    static <T> void bePositive(T[] arr){
        System.out.println("Processing Array:" + Arrays.toString(arr));
        Object[] output = new Object[arr.length];

        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) {
```



Caption (required) ✓

Describe/highlight what's being shown

The code with the necessary information for problem 3.

Explanation (required) ✓

Explain in concise steps how this logically works

EDIT RESPONSE

The bePositive method takes an array of any type, prints the array, converts each element to its positive value while keeping its original data type, stores the results in an output array, and prints the output array with each element's type. And Again it prints end of process.

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:

The most difficult part about this homework was the set up challenges I faced, which thankfully were resolved by the professor. I greatly appriciate that.

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/st278/st278-IT114-M2024/pull/2

COLLAPSE

Task #3 - Points: 1

Text: Add Screenshot of Wakatime

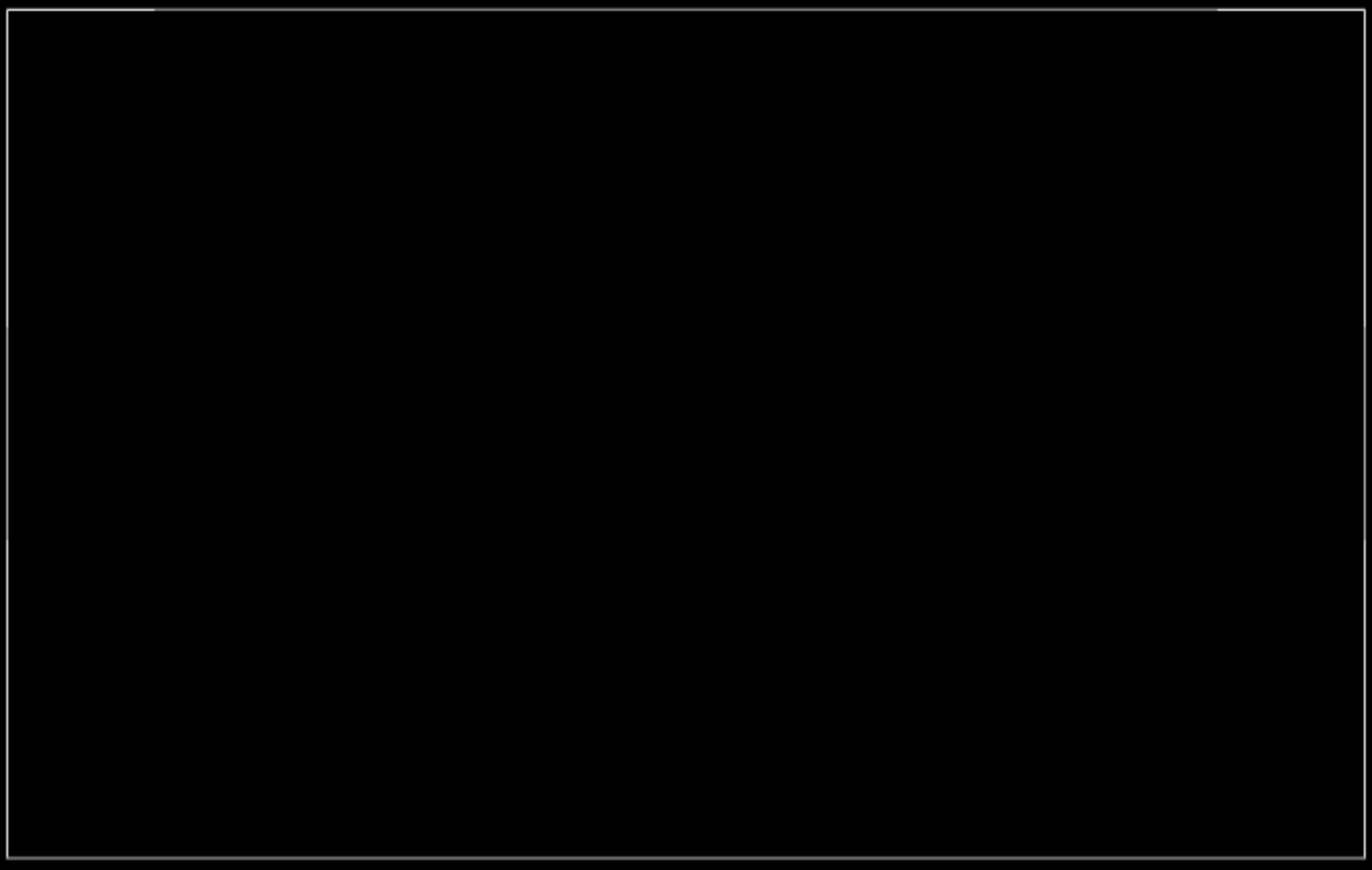
Details:

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



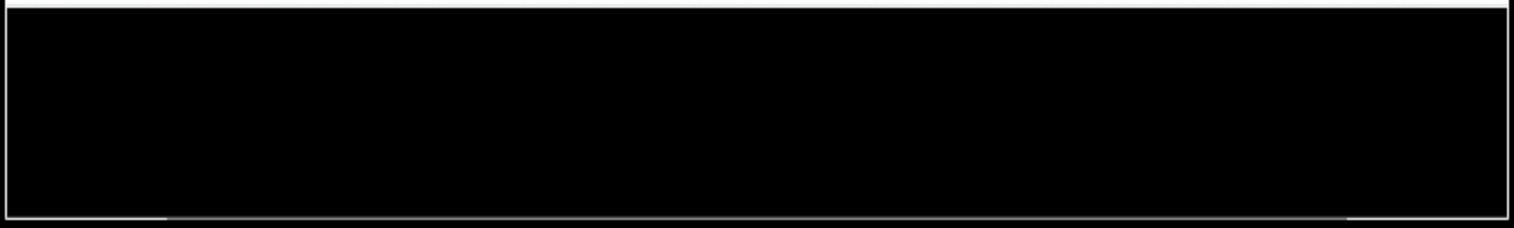
Showing the overall repository time of when I worked on It

Files

1 hr 13 mins	M2/Problem1.java
23 mins	M2/Problem2.java
15 mins	M2/Problem3.java
1 min	.gitignore
42 secs	M2
9 secs	M2/Problem2.jaav
8 secs	.git/index.lock
4 secs	README.md

Branches

1 hr 54 mins	M2-Java-Problems



Specific times on how long I worked on what.

End of Assignment