Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-002-S2024/it114-m2-java-problems/grade/pd438

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

Submissions:

Submission Selection

1 Submission [active] 3/6/2024 7:56:43 PM

Instructions

COLLAPSE

Guide:

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

Branch name: M2-Java-Problems

Branch name: M2-Java-Problems

Tasks: 8 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)

Checklist

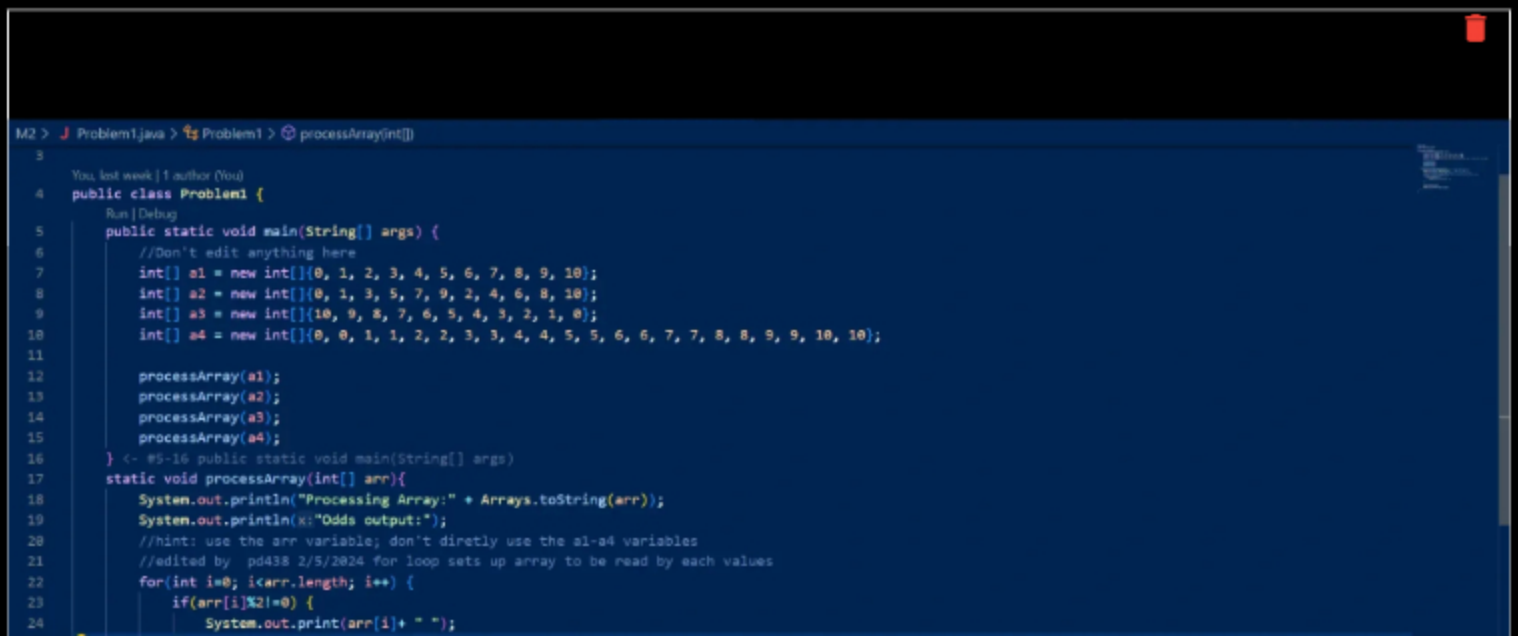
*The checkboxes are for your own tracking

#	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:

Gallery Style: Large View

Small Medium Large



M2 > Problem1.java > Problem1 > processArray(int[])

```
3
    You, last week | 1 author (You)
4   public class Problem1 {
        Run | Debug
5       public static void main(String[] args) {
6           //Don't edit anything here
7           int[] a1 = new int[]{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
8           int[] a2 = new int[]{0, 1, 3, 5, 7, 9, 2, 4, 6, 8, 10};
9           int[] a3 = new int[]{10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0};
10          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};
11
12          processArray(a1);
13          processArray(a2);
14          processArray(a3);
15          processArray(a4);
16      } <- #5-16 public static void main(String[] args)
17      static void processArray(int[] arr){
18          System.out.println("Processing Array:" + Arrays.toString(arr));
19          System.out.println("Odds output:");
20          //hint: use the arr variable; don't diretly use the a1-a4 variables
21          //edited by  pd438 2/5/2024 for loop sets up array to be read by each values
22          for(int i=0; i<arr.length; i++) {
23              if(arr[i]%2!=0) {
24                  System.out.print(arr[i]+ " ");
```

Problem 1 Code Complete.

Checklist Items (5)

#1 Edits were done only in the processArray() method and original template code/comments remain untouched

#2 Only arr is used (no direct usage of a1, a2, a3, a4)

#3 Only odd values output (not odd indexes/keys)

#4 Includes code comments with student's ucid and date

#5 Terminal output is fully visible

```
emiel@LAPTOP-B9RDLNS1 MINGW64 ~/OneDrive/Pictures/Documents/IT114/PD438_IT114_002 (main)
$ java M2/Problem1.java
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:
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
```

Problem1 Output Complete

Checklist Items (1)

#5 Terminal output is fully visible

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:

Odd Integers are determined by having a remainder that is not 0. Would do this by printing out indexes in the array that do not have a remainder of 0.

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 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)

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:

Gallery Style: Large View

Small Medium Large

```java
    static void getTotal(double[] arr) {
        System.out.println("Processing Array:" + Arrays.toString(arr));
        double total = 0;
        String totalOutput = "";

        // pd438 2/5/2024 Calculate the sum of the elements in the array
        for (int i = 0; i < arr.length; i++) {
            double a = arr[i];
            total += a;
        }

        // Format the total to a string with rounding to two decimal places
        DecimalFormat df = new DecimalFormat(pattern:"#.##");
        total = Double.valueOf(df.format(total));       You, 4 weeks ago • homework2 Complete

        // set the double to a string variable
        //  ensure rounding is to two decimal places (i.e., 0.10, 0.01, 1.00)
        total *= 100.0;
        total = Math.round(total);
        total /= 100;
        totalOutput = String.format(format:"%.2f", total) + "";

        // end add/edit section
        System.out.println("Total is " + totalOutput);
        System.out.println(x:"End process");
    } <- #18-43 static void getTotal(double[] arr)
} <- #5-44 public class Problem2
```

The changes Complete for the code.

Checklist Items (4)

#1 Edits were done only in the getTotal() method and original template code/comments remain untouched (unless noted)

#2 Only arr is used (no direct usage of a1, a2, a3, a4)

#3 Passed in array's values get summed AND rounded to two decimal places like currency (i.e., 0.00, 0.10, 1.10)

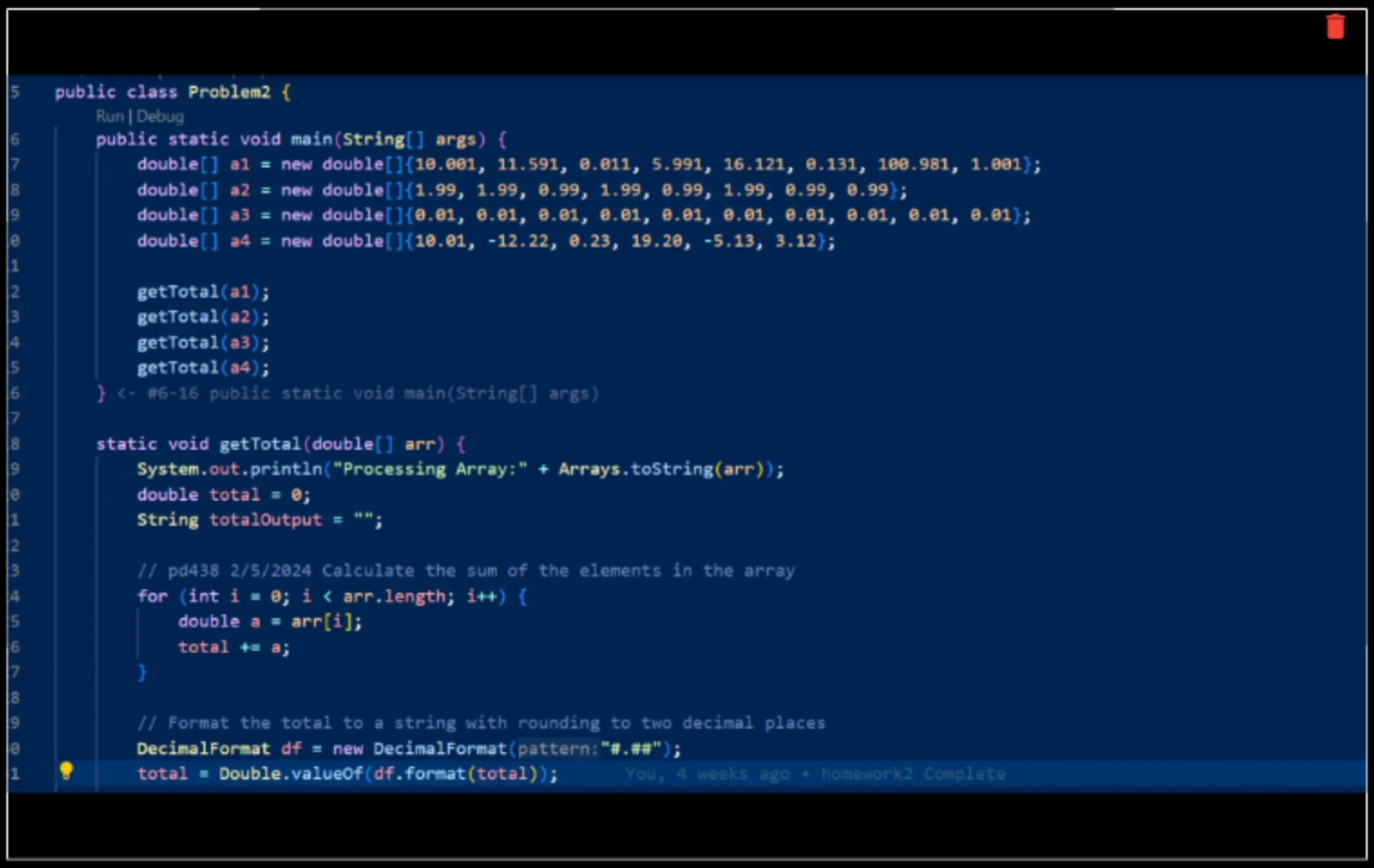
#4 Includes code comments with student's ucid and date

```
emiel@LAPTOP-B9RDLNS1 MINGW64 ~/OneDrive/Pictures/Documents/IT114/PD438_IT114_002 (main)
$ java M2/Problem2.java
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
```

Terminal Output Complete

Checklist Items (1)

#5 Terminal output is fully visible



```java
public class Problem2 {
    Run | Debug
    public static void main(String[] args) {
        double[] a1 = new double[]{10.001, 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);
    } <- #6-16 public static void main(String[] args)

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

        // pd438 2/5/2024 Calculate the sum of the elements in the array
        for (int i = 0; i < arr.length; i++) {
            double a = arr[i];
            total += a;
        }

        // Format the total to a string with rounding to two decimal places
        DecimalFormat df = new DecimalFormat(pattern:"#.##");
        total = Double.valueOf(df.format(total));       You, 4 weeks ago • homework2 Complete
```

Beginning part of the code.

Checklist Items (0)

^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:

First we add up all of the doubles in the array. Then set the total to the format requested which was 2 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)

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	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:

Gallery Style: Large View

Small Medium Large

```
static <T> void bePositive(T[] arr) {
    System.out.println("Processing Array:" + Arrays.toString(arr));

    // Create an array to store the positive values
    Object[] output = new Object[arr.length];
    //pd438 3/6/2024 created a for loop here        You, 1 second ago • Uncommitted changes
    for (int i = 0; i < arr.length; i++) {
        if (arr[i] instanceof Number) {
            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) {
                output[i] = Math.abs(Integer.parseInt((String) arr[i]));
                output[i] = arr[i];
        } else {
            output[i] = arr[i];
        }
    } <- #23-36 for (int i = 0; i < arr.length; i++)

    StringBuilder sb = new StringBuilder();
    for (Object i : output) {
```

```
            if (sb.length() > 0) {
                sb.append(str:",");
            }
            sb.append(String.format(format:"%s (%s)", i, i.getClass().getSimpleName().substring(beginIndex:0, endIndex:1)));
        } <- #39-44 for (Object i : output)
        System.out.println("Result: " + sb.toString());
    } <- #17-46 static <T> void bePositive(T[] arr)
<- #4-47 public class Problem3
```

Code updated with comment to display change.

Checklist Items (4)

#1 Edits were done only in the bePositive() method and original template code/comments remain untouched

#2 Only arr is used (no direct usage of a1, a2, a3, a4)

#3 Passed in array's values will get converted to a positive version AND converted back to the original data type

#4 Includes code comments with student's ucid and date

```
emiel@LAPTOP-B9RDLNS1 MINGW64 ~/OneDrive/Pictures/Documents/IT114/PD438_IT114_002 (main)
$ java M2/Problem3.java
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)
```

Terminal Output.

Checklist Items (1)

#5 Terminal output is fully visible

COLLAPSE

Task #2 - Points: 1

Text: Explain your solution

Checklist *The checkbox 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:

If array contained numbers or doubles, we set the numbers to positive using the math.abs function. If the array contained strings we convert it into integers then used the math.abs function.

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.

Response:

This took alot of time and many of the concepts were confusing for me. And the information was not enough for me to comprehend looking at it now. I wish there were more sources to see more material.

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/PD438/PD438_IT114_002/pull/9

End of Assignment