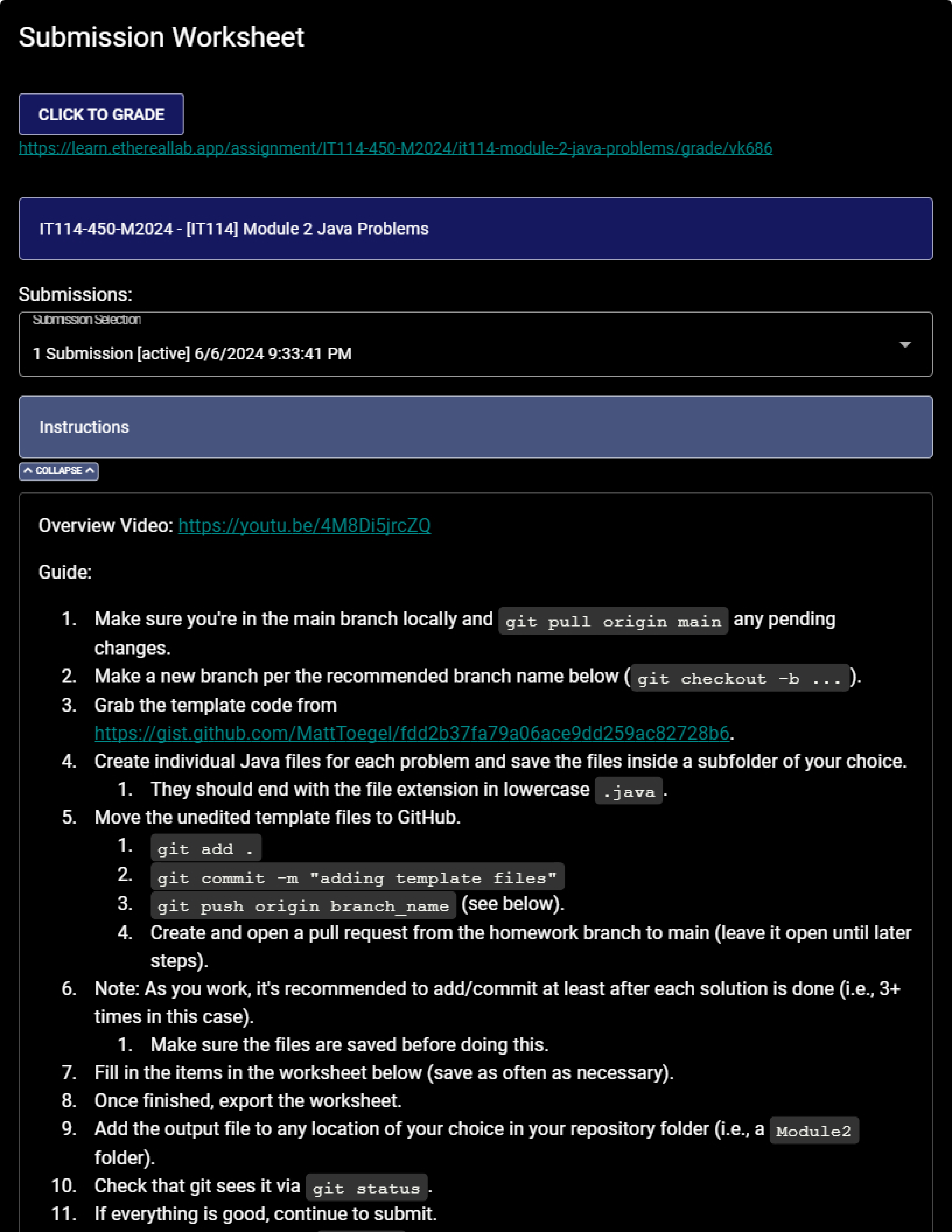


Submission Worksheet

https://learn.ethereallab.app/assignment/IT114-450-M2024/it114-module-2-java-problems/grade/vk686

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

Submissions:

Submission Selection

1 Submission [active] 6/6/2024 9:33:41 PM

Instructions

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

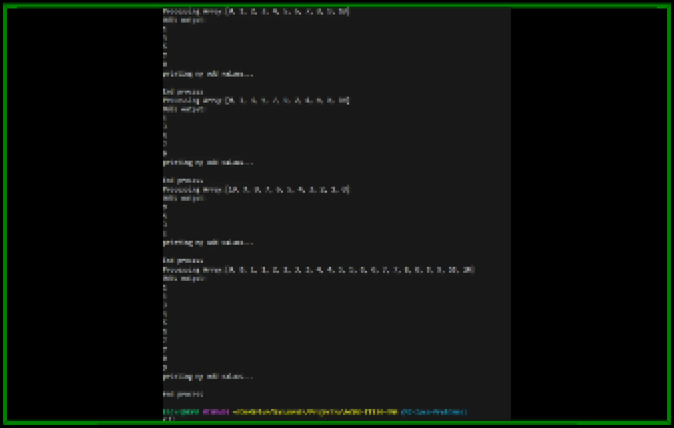
Text: Screenshot of the Problem 1 Solved Code and Output

ⓘ 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

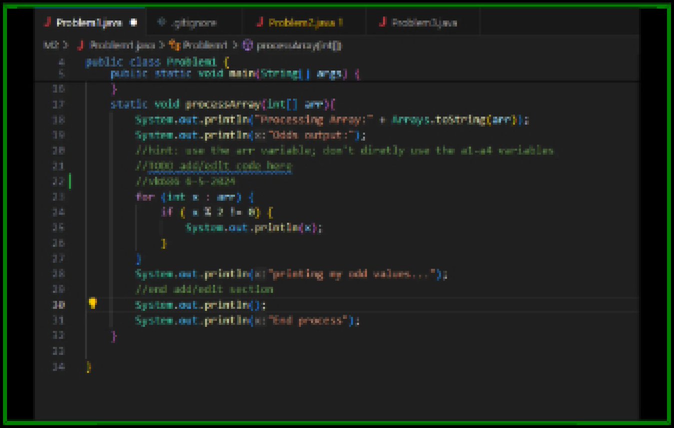


```
Processing Array:[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Odds output:
1
3
5
7
9
printing my odd values...

End process
Processing Array:[[illegible]]
Odds output:
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
printing my odd values...

End process
Processing Array:[[illegible]]
Odds output:
[illegible]
[illegible]
[illegible]
[illegible]
printing my odd values...

End process
Processing Array:[[illegible]]
Odds output:
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
printing my odd values...

End process
```

Caption (required) ✓

Describe/highlight what's being shown

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



```java
public class Problem1 {
    public static void main(String[] args) {
    }
    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
        //TODO add/edit code here
        //[illegible] 6-5-2024
        for (int x : arr) {
            if ( x % 2 != 0) {
                System.out.println(x);
            }
        }
        System.out.println(x:"printing my odd values...");
        //end add/edit section
        System.out.println();
        System.out.println(x:"End process");
    }

}
```

Caption (required) ✓

Describe/highlight what's being shown

The output of the solved problem

The screenshot of my solution code

Explanation (required) ✓

Explain in concise steps how this logically works

PREVIEW RESPONSE

It implements a for-each loop that goes through each item in the array and uses num % 2!= 0 to check if the item is odd.

● ^ COLLAPSE ^

Problem 2 (3 pts.)

● ^ COLLAPSE ^

Task #1 - Points: 1

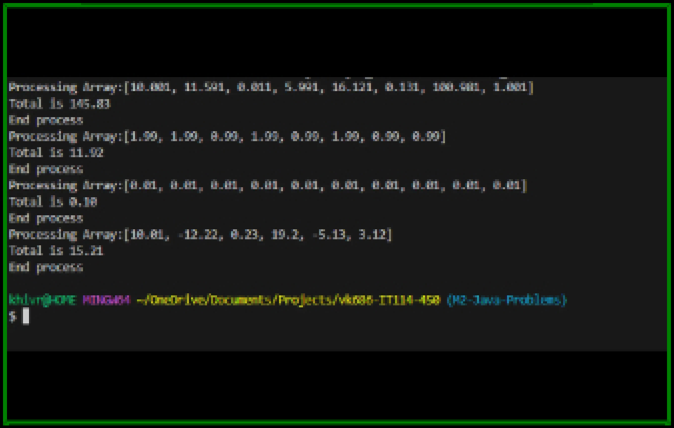
Text: Screenshot of the Problem 2 Solved Code and Output

Details:

Only make edits where the template code mentions.

Solution should ensure that any passed in array will have its values 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



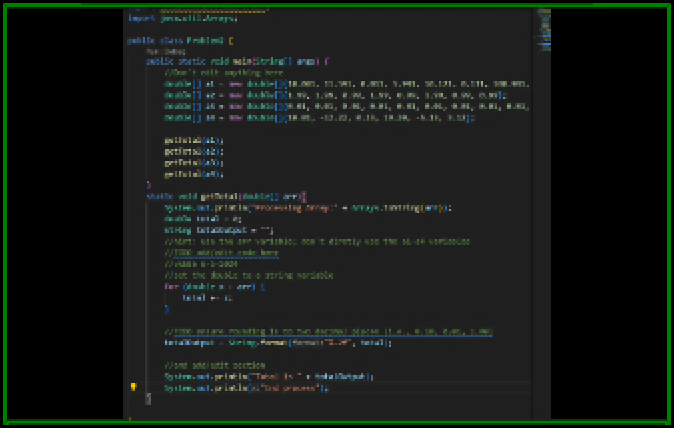
```
Processing Array:[10.001, 11.591, 0.011, 5.991, 16.121, 0.131, 100.981, 1.001]
Total is 145.83
End process
Processing Array:[1.99, 1.99, 0.99, 1.99, 0.99, 1.99, 0.99, 0.99]
Total is 11.92
End process
Processing Array:[0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01]
Total is 0.10
End process
Processing Array:[10.01, -12.22, 0.23, 19.2, -5.13, 3.12]
Total is 15.21
End process

khlvr@HOME MINGW64 ~/OneDrive/Documents/Projects/vk686-IT114-450 (M2-Java-Problems)
$ 
```

Caption (required) ✓

Describe/highlight what's being shown

Screenshot the output of the solved problem

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



```
import java.util.Arrays;

public class Problem2 {
    [illegible]
    public static void main(String[] args) {
        //Don't edit anything here
        double[] a1 = new double[]{10.001, 11.591, 0.011, 5.991, 16.121, 0.131, 100.981,
        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,
        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;
        String totalOutput = "";
        //hint: use the arr variable; don't directly use the a1-a4 variables
        //TODO add/edit code here
        //vk686 [illegible]
        //set the double to a string variable
        for (double i : arr) {
            total += i;
        }

        //TODO ensure rounding is to two decimal places (i.e., 0.10, 0.01, 1.00)
        totalOutput = String.format(format:"%.2f", total);

        //end add/edit section
        System.out.println("Total is " + totalOutput);
        System.out.println(x:"End process");
    }
```

Caption (required) ✓

Describe/highlight what's being shown

Screenshot the code solution

Explanation (required) ✓

Explain in concise steps how this logically works

PREVIEW RESPONSE

I used "for" loop to sum up all the elements in the array. String.format("%.2f", total) -to round the total to two decimal places

Problem 3 (3 pts.)

COLLAPSE

COLLAPSE

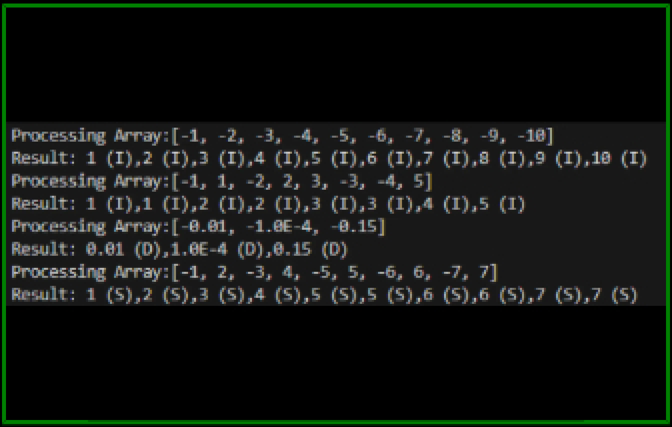
Task #1 - Points: 1

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



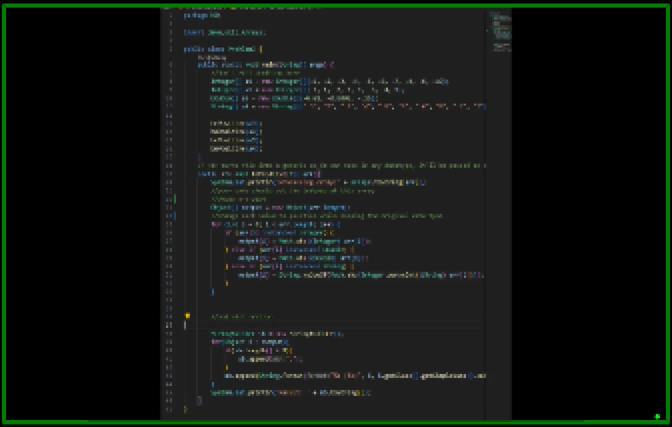
```
Processing Array:[-1, -2, -3, -4, -5, -6, -7, -8, -9, -10]
Result: 1 (I),2 (I),3 (I),4 (I),5 (I),6 (I),7 (I),8 (I),9 (I),10 (I)
Processing Array:[-1, 1, -2, 2, 3, -3, -4, 5]
Result: 1 (I),1 (I),2 (I),2 (I),3 (I),3 (I),4 (I),5 (I)
Processing Array:[-0.01, -1.0E-4, -0.15]
Result: 0.01 (D),1.0E-4 (D),0.15 (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),7 (S),7 (S)
```

Caption (required) ✓

Describe/highlight what's being shown

The output of the solved problem

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



Caption (required) ✓

Describe/highlight what's being shown

The screenshot of the code solution

Explanation (required) ✓

Explain in concise steps how this logically works

PREVIEW RESPONSE

It runs through the section once, checks each value in the area against different types, and then uses each value's absolute value. By using "instanceof," the computer finds the original datatype, converts it to that original datatype (because you can't use methods that original datatype (because you can't use methods that belong to an object through a general object reference no matter what type the object is), and then gets the absolute value. When the object is a String, it is turned into a String, parsed into a number, and its absolute value is taken. It is then turned back into a String. In the end, the outputs for all of these absolute numbers are added to the "output" array.

● 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:

Dicht have any issues, but I learned how to check for odds using "num % 2!= 0", I learned how to round per the requirements via "String.format("%.2f", total)", and I learned about casting in problem 3.

● ^ 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/VK686NJ/vk686-IT114-450/pull/3

● ^ COLLAPSE ^

Task #3 - Points: 1

Text: Add Screenshot of Wakatime

Details:

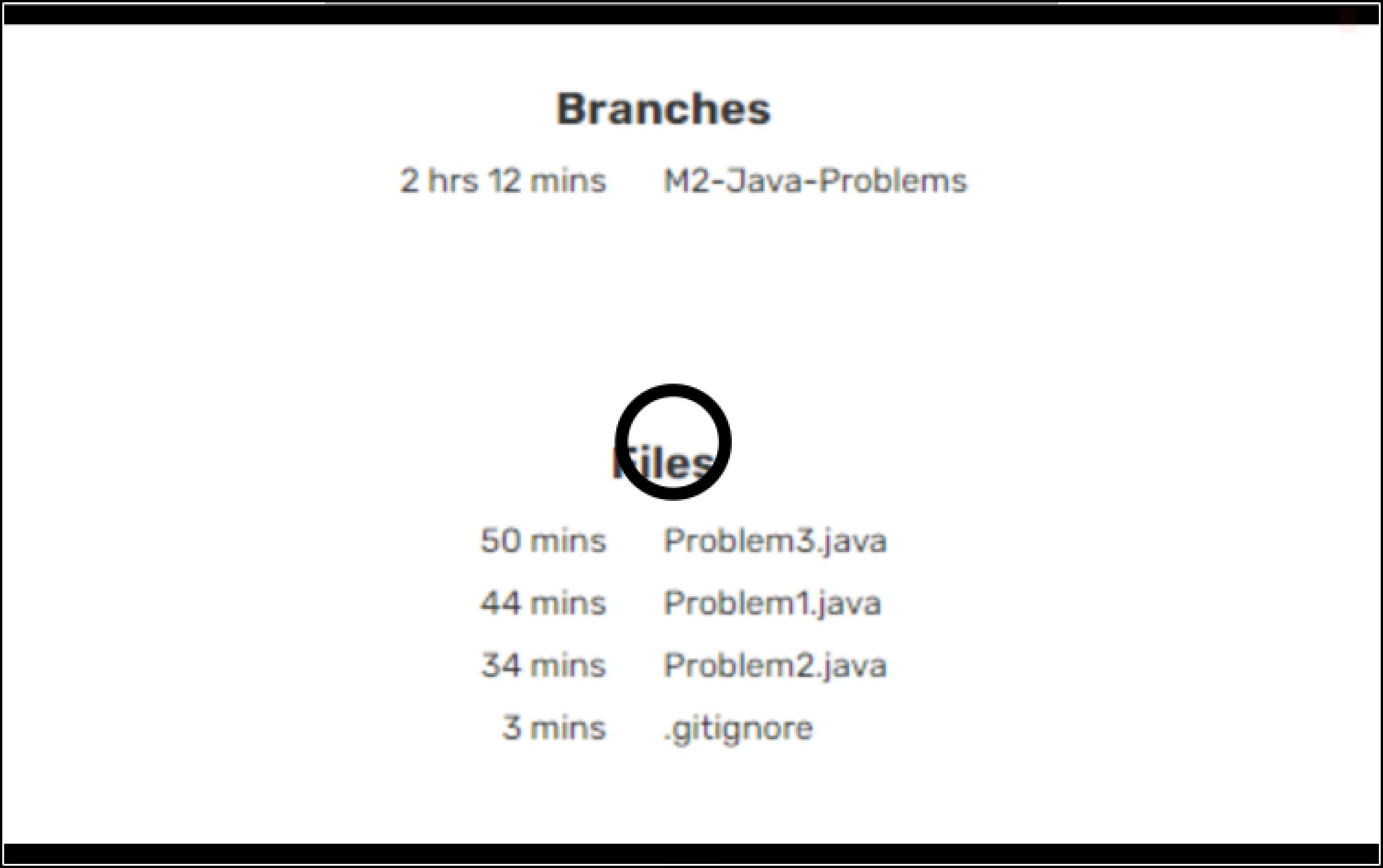
Details:

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



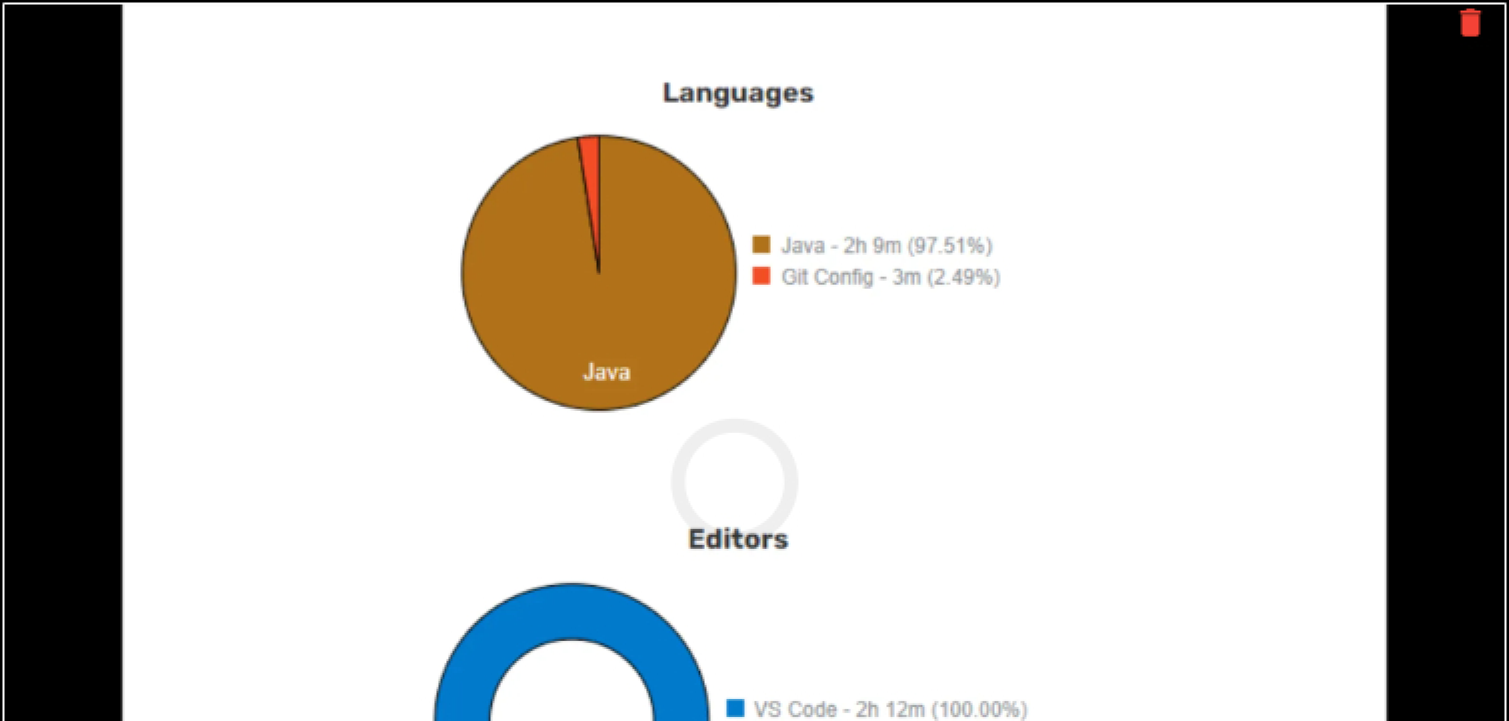
Branches

2 hrs 12 mins	M2-Java-Problems

Files

50 mins	Problem3.java
44 mins	Problem1.java
34 mins	Problem2.java
3 mins	.gitignore

WakaTime



WakaTime

End of Assignment