Coding Assignment 1

Due: Oct. 7, 23:59 PT

Overview

This assignment includes 3 parts, covering Lesson 3 - Lesson 11. Unless specified, you can assume the inputs are always correct, so no input validation is needed. Please follow the specifications (in red) and make sure there is no compiler error in your code.

You will receive a zero if...

- Your code is not runnable (wrong class/method name, wrong parameter & return type, compiler errors, etc.)
- Your code is from the Internet, AI, or other student.

Setup

1. Create an empty project in IntelliJ (or other IDE), project name doesn't matter.
2. Create a package called "assignment1"
3. Create 3 classes under the assignment1 package: "MyTime", "MyCard", and "MyMatrix"
4. (Optional) Create a Main class with `psvm` to test your code (don't need to submit)

Part 1. Time Difference (3.5 pts)

In "MyTime", write a public static method called "printTimeDifference" that takes two Strings as two times in military format (hhmm) and prints the number of hours and minutes between the two times. If the first time is later than the second time, assume the second time is the next day (+24 hours).

The output format should be "X hour(s) Y minute(s)" where X is the hour difference and Y is the minute difference (other parts should be fixed). No requirement for line breaker.

Additional requirements

- Any repeated procedure should be in a separate method.
- Simplify the logic as much as possible (you don't need more than two "if").
- Use formatted String for output (i.e. use `System.out.printf` or `String.format`).

Examples

- Example 1: The first time smaller than the second time

  Code: `printTimeDifference("0120", "1510");`
  Output: 13 hour(s) 50 minute(s)

- Example 2: The first time greater than the second time

  Code: `printTimeDifference("1635", "0250");`
  Output: 10 hour(s) 15 minute(s)

Part 2. Card Suit (3.5 pts)

In “MyCard”, implement...

- A constructor takes a String, which is the shorthand of a card. The format is the shorthand for the rank followed by the shorthand of the suit.
- A getSuit method takes nothing but returns a String for the full term of the suit. If the shorthand not valid, return “Unknown”.

Here’s a table for valid shorthand and the corresponding full terms. The ranks are for reference.

	Shorthand	**Full**
Ranks	A	Ace
	2 - 10	Card values (Two for 2, Three for 3, etc.)
	J	Jack
	Q	Queen
	K	King
Suits	D	Diamonds
	H	Hearts
	S	Spades
	C	Clubs

Additional requirements

- Any instance variable should be private.
- Use switch statement.
- No additional output.
- Consider efficiency – should the logic be in the constructor, or in the method?

Examples

- Example 1: Shorthand = 10S, rank = “Ten”, suit = “Spades”

  ```
  MyCard card = new Card("10S")
  String suit = card.getSuit();
  ```

  After executing the code, `suit` should be "Spades".

- Example 2: Shorthand = QC, rank = “Queen”, suit = “Clubs”

  ```
  MyCard card = new Card("QC")
  String suit = card.getSuit();
  ```

  After executing the code, `suit` should be "Clubs".

- Example 3: Shorthand = AJ, rank = “Ace”, suit = “Unknown”

  ```
  MyCard card = new Card("11H")
  String suit = card.getSuit();
  ```

  After executing the code, `suit` should be "Unknown".

Part 3. Matrix Multiplication (3 pts)

In “MyMatrix”, implement...

- A constructor takes a 2-d int array.
- A getMatrix method to get the 2-d int array that is used to construct the Matrix.
- A multiply method, which takes another MyMatrix object, computes the matrix multiplication (between the current MyMatrix object and the parameter MyMatrix object), and returns a MyMatrix object for the resulting matrix.

Additional requirements

- Any instance variable should be private.
- No additional output.
- The underlying 2-d int array cannot be changed.

Examples

- Example 1: the underlying 2-d int array cannot be changed

  ```
  int[][] array1 = {{2, 4}, {3, 4}};
  MyMatrix matrix1 = new MyMatrix(array1);
  array1[0][0] = 5;
  ```

  After executing the code, `matrix1.getMatrix()` should still return `{{2, 4}, {3, 4}}`.

- Example 2: calculating $\begin{bmatrix} 2 & 4 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 1 & 2 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 6 & 16 \\ 7 & 18 \end{bmatrix}$

  ```
  int[][] array2 = {{1, 2}, {1, 3}};
  MyMatrix matrix2 = new MyMatrix(array2);
  MyMatrix resultMatrix = matrix1.multiply(matrix2);
  int[][] resultArray = resultMatrix.getMatrix();
  ```

  After executing the code, `resultArray` should be `{{6, 16}, {7, 18}}`.
  The example shows multiplication between two 2 x 2 matrix, but your method should work on any correct sizes of matrices.

Submission & Grading

Submit MyTime.java, MyCard.java, and MyMatrix.java to the corresponding Canvas assignment (Drag the files into the submission box and click “Upload”).

You can submit as many times as you want to pass the auto-graded tests. No points back for any errors that are caught by the auto-grader.

After the assignment is due, the grader will manually check your code. For each part, you will get up to 1.5 pts deducted if your code doesn’t satisfy the additional requirements. That is, your final score on the assignment may be lower than the auto-grading result but will not be higher.