

CHRIST (Deemed to be University)
Department of Computer Science
Master of Artificial Intelligence and Machine Learning

Course: MAI271 – JAVA Programming

Exercise No: LAB Exercise – 5

Date: 15 – 11 – 2024

Duration: 2 Hrs

Question 1 (4 Marks)

Create an interface named 'BankInterface' containing abstract methods 'getBalance' and 'getInterestRate'. Deposit amounts of 10000, 150000, and 200000 into banks A, B, and C, respectively. 'BankA', 'BankB', and 'BankC' implement the 'BankInterface' interface, each defining methods 'getBalance' and 'getInterestRate'. Invoke these methods by instantiating objects of each class.

BankA provides an interest rate of 7% based on the balance.

BankB offers an interest rate of 7.4% based on the balance.

BankC provides an interest rate of 7.9% based on the balance.

Display the balance and interest rate of each bank separately.

Question 2 (5 Marks)

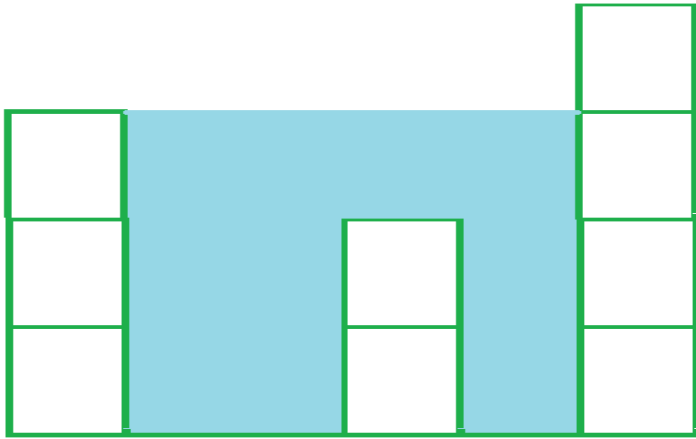
Imagine a cityscape with an array of blocks, each represented by a non-negative integer indicating its height. During the rainy season, the city faces a unique water conservation challenge. If the width of each block is considered to be 1 unit, we aim to determine the volume of water that can be conserved between these blocks.

As an urban planner, your task is to develop an innovative water conservation system. This involves creating an interface named **WaterConservationSystem** with a method **calculateTrappedWater(int[] blockHeights)**. This method should efficiently calculate and return the total volume of water that can be conserved between the city blocks.

To implement this system, design an abstract class called **RainySeasonConservation** that implements the **WaterConservationSystem** interface. This abstract class will serve as a foundation for specific implementations based on various block configurations.

Subsequently, create a class named **CityBlockConservation** that extends **RainySeasonConservation**. Implement the **calculateTrappedWater(int[] blockHeights)** method to address the given block heights, represented by the array **blockHeights[]**.

Test Case 1:



Bars for input {3, 0, 0, 2, 0, 4}
Total trapped water = 3 + 3 + 1 + 3 = 10

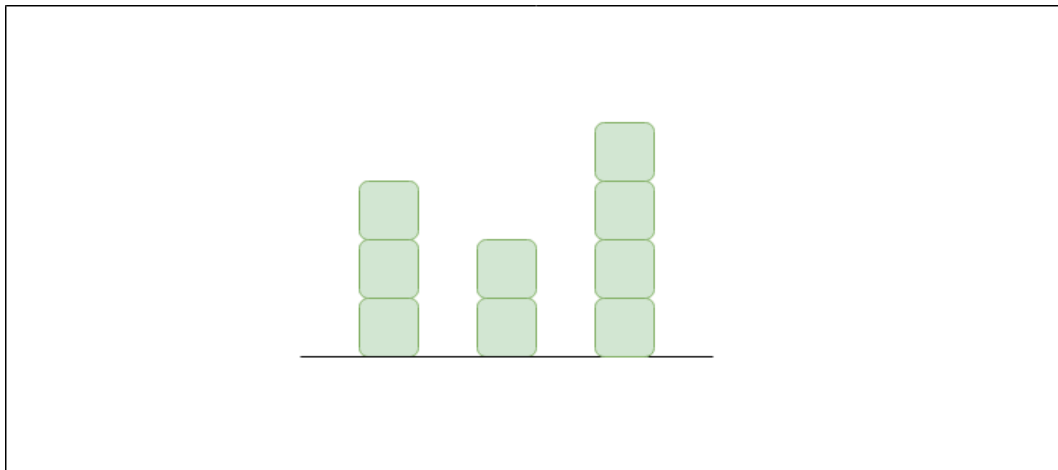
Test Case 2:

Input: `arr[] = {3, 0, 2, 0, 4}`

Output: 7

Explanation: Structure is like below.

We can trap “3 units” of water between 3 and 2,
“1 unit” on top of bar 2 and “3 units” between 2 and 4.



General Instruction:

1. Ensure that your code includes relevant comments to enhance readability and understanding. Subsequently, upload your code to GitHub for version control and collaborative access.
2. Include descriptive comments within the code, explaining its functionality and logic.

3. In the Google Classroom submission, include the GitHub URL where your code is hosted.
4. Attach a PDF document named "your_register_number_exercise_No.pdf" to the submission. The PDF document should include screenshots of the code and the output screen.
5. Upload the answer document & GitHub URL in Google Classroom on or before the deadline mentioned. Evaluation will not be considered for late submission