

Java Lecture > src > Lab05 > J BankInterface.java > ...

```
1  package Lab05;
2  public interface BankInterface {
3
4      double getBalance();
5
6      double getInterestRate();
7  }
8
9  class BankA implements BankInterface {
10
11      private final double balance = 10000; // Initial balance for Bank A
12
13      @Override
14      public double getBalance() {
15          |   return balance;
16      }
17
18      @Override
19      public double getInterestRate() {
20          |   return 7.0;
21      }
22  }
23
24  class BankB implements BankInterface {
25
26      private final double balance = 150000; // Initial balance for Bank B
27
28      @Override
29      public double getBalance() {
30          |   return balance;
31      }
32
33      @Override
34      public double getInterestRate() {
35          |   return 7.4;
36      }
37  }
38
```

```

39 class BankC implements BankInterface {
40
41     private final double balance = 200000; // Initial balance for Bank C
42
43     @Override
44     public double getBalance() {
45         return balance;
46     }
47
48     @Override
49     public double getInterestRate() {
50         return 7.9;
51     }
52 }
53
54 class Main {
55
56     Run | Debug
57     public static void main(String[] args) {
58
59         // Create bank instances with pre-defined initial balances
60         BankInterface bankA = new BankA();
61         BankInterface bankB = new BankB();
62         BankInterface bankC = new BankC();
63
64         // Print information for each bank
65         System.out.println(x:"Bank A");
66         System.out.println("Balance:" + bankA.getBalance());
67         System.out.println("Interest Rate: " + bankA.getInterestRate() + "%");
68         System.out.println();
69
70         System.out.println(x:"Bank B");
71         System.out.println("Balance:" + bankB.getBalance());
72         System.out.println("Interest Rate: " + bankB.getInterestRate() + "%");
73         System.out.println();
74
75         System.out.println(x:"Bank C");
76         System.out.println("Balance:" + bankC.getBalance());
77         System.out.println("Interest Rate: " + bankC.getInterestRate() + "%");
78     }
79 }

```

```

ing\Code\User\workspaceStorage\2c303dbc6f9f4
Bank A
Balance:10000.0
Interest Rate: 7.0%

Bank B
Balance:150000.0
Interest Rate: 7.4%

Bank C
Balance:200000.0
Interest Rate: 7.9%
PS D:\Java Projects>

```

Java Lecture > src > Lab05b > WaterConservationSystem.java > CityBlockConservation

```
1  package Lab05b;
2  // WaterConservationSystem interface
3  public interface WaterConservationSystem {
4      int calculateTrappedWater(int[] blockHeights);
5  }
6
7  // RainySeasonConservation abstract class implementing WaterConservationSystem interface
8  abstract class RainySeasonConservation implements WaterConservationSystem {
9
10 }
11
12 // CityBlockConservation class extending RainySeasonConservation
13 class CityBlockConservation extends RainySeasonConservation {
14
15     @Override
16     public int calculateTrappedWater(int[] blockHeights) {
17         if (blockHeights == null || blockHeights.length <= 2) {
18             return 0; // No trapped water if there are less than 3 blocks
19         }
20
21         int n = blockHeights.length;
22         int[] leftMax = new int[n];
23         int[] rightMax = new int[n];
24
25         // Calculate the maximum height to the left of each block
26         leftMax[0] = blockHeights[0];
27         for (int i = 1; i < n; i++) {
28             leftMax[i] = Math.max(leftMax[i - 1], blockHeights[i]);
29         }
30
31         // Calculate the maximum height to the right of each block
32         rightMax[n - 1] = blockHeights[n - 1];
33         for (int i = n - 2; i >= 0; i--) {
34             rightMax[i] = Math.max(rightMax[i + 1], blockHeights[i]);
35         }
36
37         // Calculate trapped water for each block
38         int trappedWater = 0;
39         for (int i = 0; i < n; i++) {
40             trappedWater += Math.max(0, Math.min(leftMax[i], rightMax[i]) - blockHeights[i]);
41         }
42
43         return trappedWater;
44     }
45 }
```

```

47 }
48
49 // Main class for testing
50 class Main {
51     Run | Debug
52     public static void main(String[] args) {
53         // Test Case 1
54         int[] heights1 = {3, 0, 0, 2, 0, 4};
55         WaterConservationSystem conservationSystem1 = new CityBlockConservation();
56         int result1 = conservationSystem1.calculateTrappedWater(heights1);
57         System.out.println("Test Case 1 Result: " + result1); // Expected output: 10
58
59         // Test Case 2
60         int[] heights2 = {3, 0, 2, 0, 4};
61         WaterConservationSystem conservationSystem2 = new CityBlockConservation();
62         int result2 = conservationSystem2.calculateTrappedWater(heights2);
63         System.out.println("Test Case 2 Result: " + result2); // Expected output: 7
64     }
65 }

```

PROBLEMS 26 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS D:\Java Projects> d:.; cd 'd:\Java Projects'; & 'C:\Program Files\Java\jdk-17.0.1\bin\java.exe' '-agent
6' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\SATYAM\AppData\Roaming\Code\User\workspaceSto
_1c7c2edf\bin' 'Lab05b.Main'
Test Case 1 Result: 10
Test Case 2 Result: 7
PS D:\Java Projects> 

```