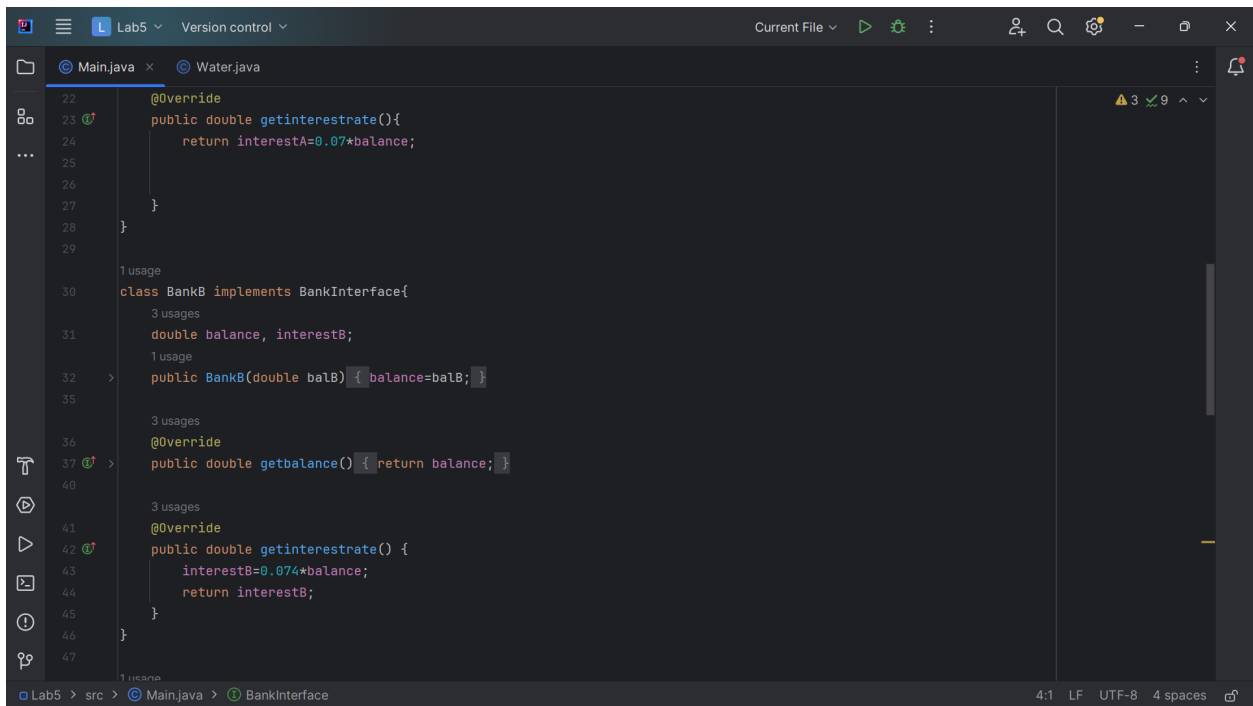


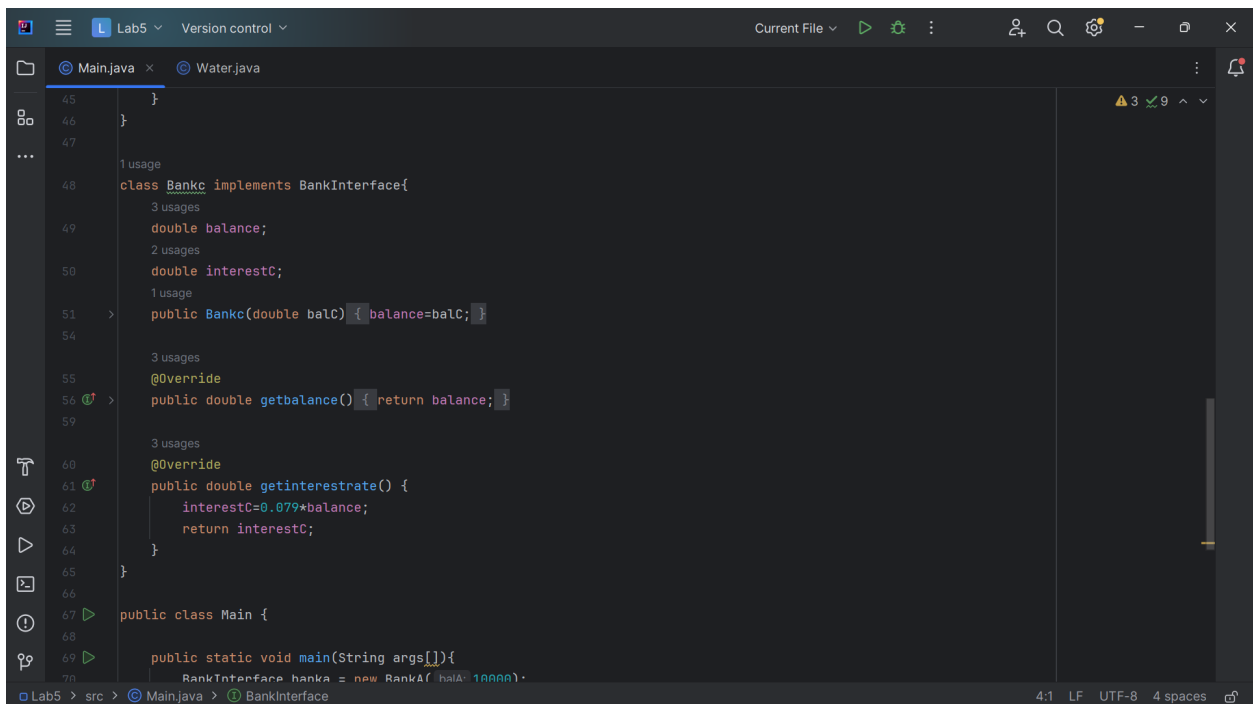
QUESTION 1



This screenshot shows an IDE window with two tabs: Main.java and Water.java. The Main.java file is open, displaying the implementation of the BankB class. The code includes an @Override method getinterestrate() and a constructor BankB. The class implements the BankInterface. The IDE interface includes a sidebar with file explorer, a top bar with version control, and a bottom status bar showing file encoding and line numbers.

```
22 @Override
23 public double getinterestrate(){
24     return interestA=0.07*balance;
25 }
26
27 }
28 }
29
30 1 usage
31 class BankB implements BankInterface{
32     3 usages
33     double balance, interestB;
34     1 usage
35     public BankB(double balB) { balance=balB; }
36
37     3 usages
38     @Override
39     public double getbalance() { return balance; }
40
41     3 usages
42     @Override
43     public double getinterestrate() {
44         interestB=0.074*balance;
45         return interestB;
46     }
47 }
```

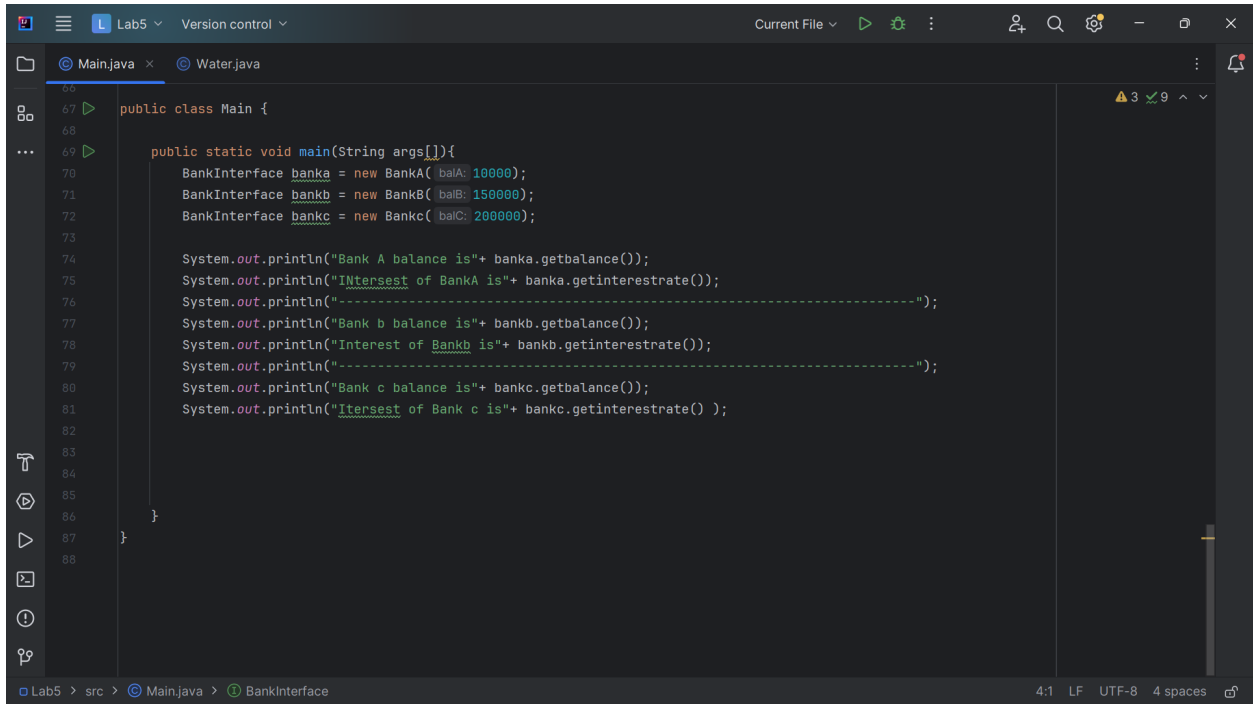
Lab5 > src > Main.java > BankInterface 4:1 LF UTF-8 4 spaces



This screenshot shows the same IDE window, but now displaying the implementation of the BankC class and the Main class. The BankC class implements the BankInterface with a different interest rate. The Main class contains a main method that creates an instance of BankA. The IDE interface is consistent with the previous screenshot.

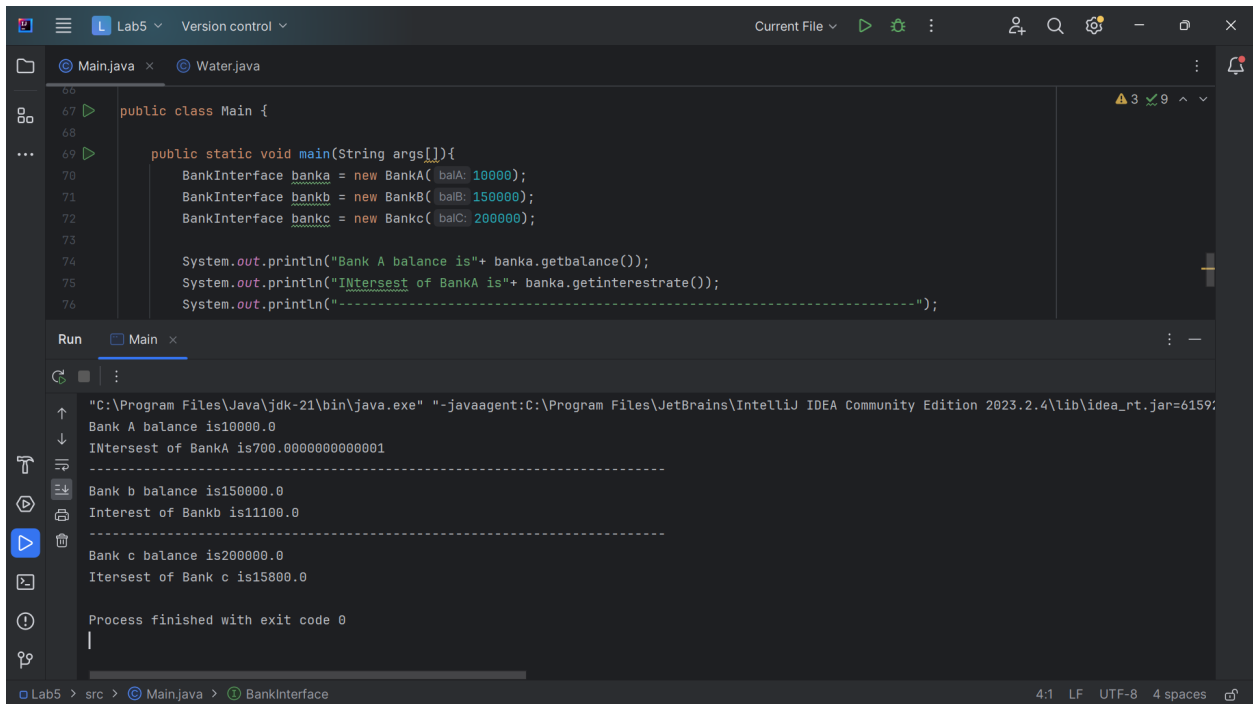
```
45 }
46 }
47
48 1 usage
49 class BankC implements BankInterface{
50     3 usages
51     double balance;
52     2 usages
53     double interestC;
54     1 usage
55     public BankC(double balC) { balance=balC; }
56
57     3 usages
58     @Override
59     public double getbalance() { return balance; }
60
61     3 usages
62     @Override
63     public double getinterestrate() {
64         interestC=0.079*balance;
65         return interestC;
66     }
67 }
68
69 public class Main {
70
71     public static void main(String args[]){
72         BankInterface bankA = new BankA( "haIA" 10000);
73     }
74 }
```

Lab5 > src > Main.java > BankInterface 4:1 LF UTF-8 4 spaces



```
66
67 public class Main {
68
69     public static void main(String args[]){
70         BankInterface banka = new BankA( balA: 10000);
71         BankInterface bankb = new BankB( balB: 150000);
72         BankInterface bankc = new BankC( balC: 200000);
73
74         System.out.println("Bank A balance is"+ banka.getbalance());
75         System.out.println("Interest of BankA is"+ banka.getinterestrate());
76         System.out.println("-----");
77         System.out.println("Bank b balance is"+ bankb.getbalance());
78         System.out.println("Interest of Bankb is"+ bankb.getinterestrate());
79         System.out.println("-----");
80         System.out.println("Bank c balance is"+ bankc.getbalance());
81         System.out.println("Interest of Bank c is"+ bankc.getinterestrate() );
82
83     }
84
85 }
86
87
88
```

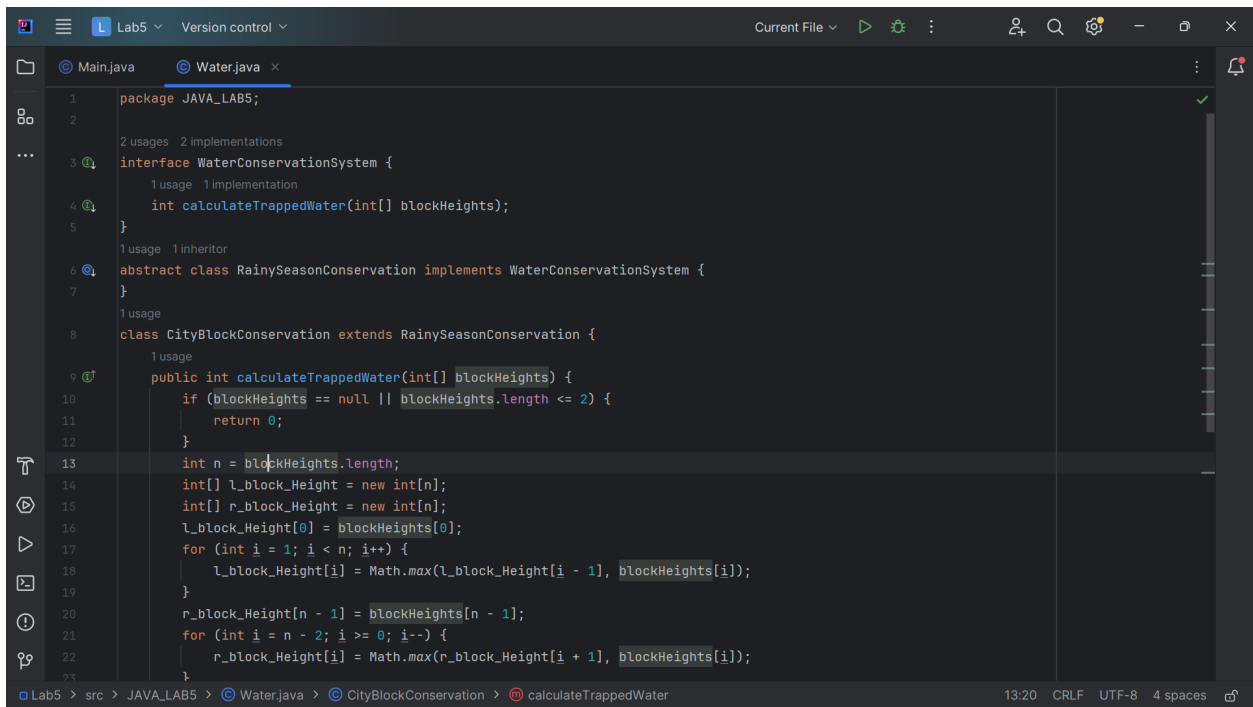
OUTPUT:



```
Run Main x
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.2.4\lib\idea_rt.jar=6159%
Bank A balance is10000.0
Interest of BankA is700.0000000000001
-----
Bank b balance is150000.0
Interest of Bankb is11100.0
-----
Bank c balance is200000.0
Interest of Bank c is15000.0

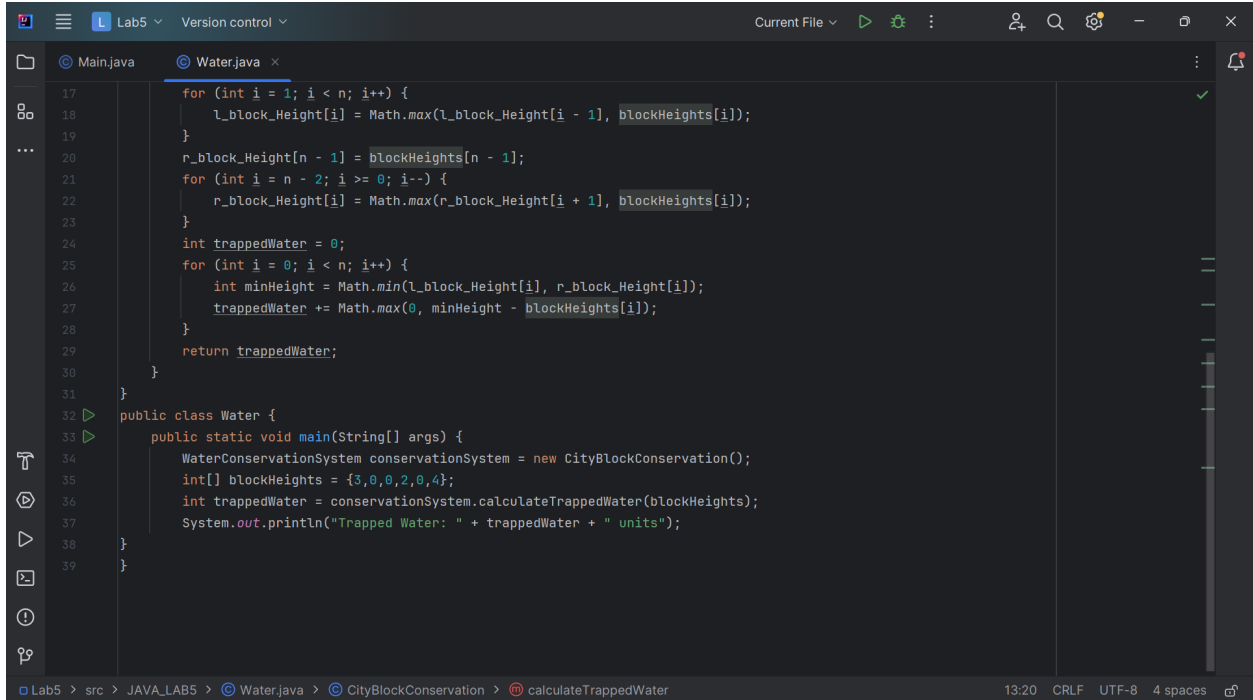
Process finished with exit code 0
```

QUESTION 2



```
1 package JAVA_LABS;
2
3 2 usages 2 implementations
4 1 usage 1 implementation
5 interface WaterConservationSystem {
6     int calculateTrappedWater(int[] blockHeights);
7 }
8 1 usage 1 inheritor
9 abstract class RainySeasonConservation implements WaterConservationSystem {
10 }
11 1 usage
12 class CityBlockConservation extends RainySeasonConservation {
13     1 usage
14     public int calculateTrappedWater(int[] blockHeights) {
15         if (blockHeights == null || blockHeights.length <= 2) {
16             return 0;
17         }
18         int n = blockHeights.length;
19         int[] l_block_Height = new int[n];
20         int[] r_block_Height = new int[n];
21         l_block_Height[0] = blockHeights[0];
22         for (int i = 1; i < n; i++) {
23             l_block_Height[i] = Math.max(l_block_Height[i - 1], blockHeights[i]);
24         }
25         r_block_Height[n - 1] = blockHeights[n - 1];
26         for (int i = n - 2; i >= 0; i--) {
27             r_block_Height[i] = Math.max(r_block_Height[i + 1], blockHeights[i]);
28         }
29     }
30 }
```

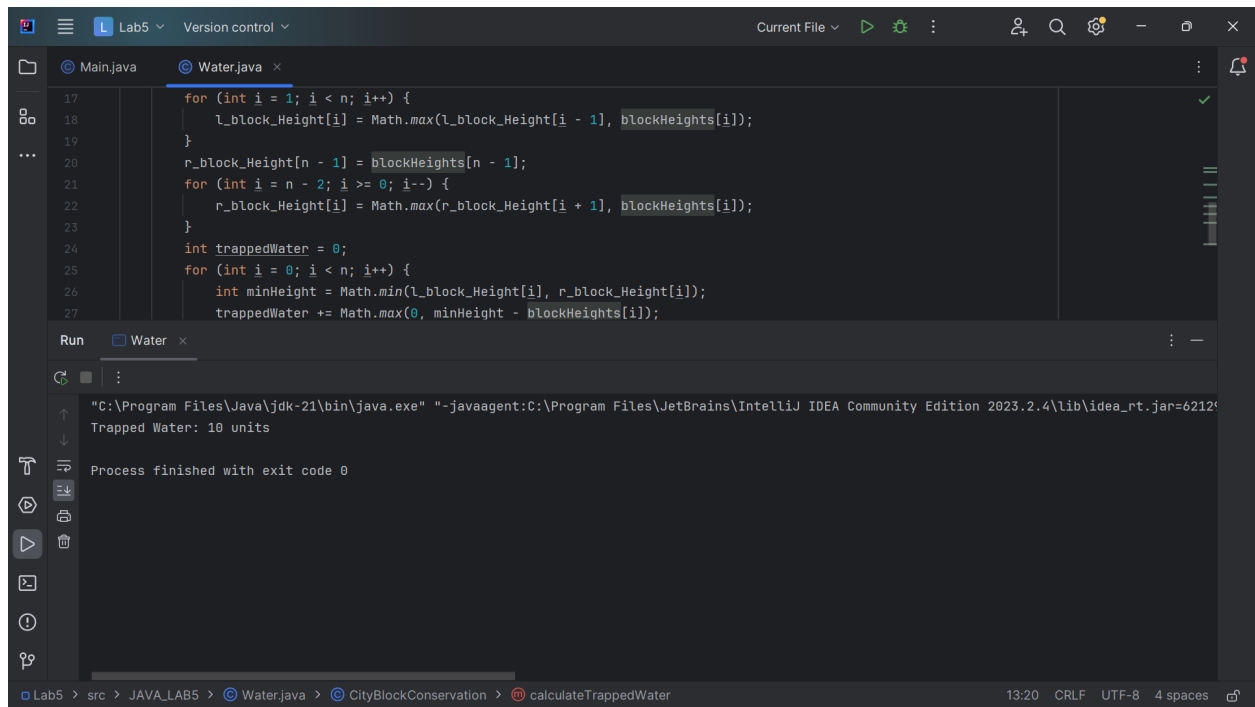
Lab5 > src > JAVA_LABS > Water.java > CityBlockConservation > calculateTrappedWater 13:20 CRLF UTF-8 4 spaces



```
17         for (int i = 1; i < n; i++) {
18             l_block_Height[i] = Math.max(l_block_Height[i - 1], blockHeights[i]);
19         }
20         r_block_Height[n - 1] = blockHeights[n - 1];
21         for (int i = n - 2; i >= 0; i--) {
22             r_block_Height[i] = Math.max(r_block_Height[i + 1], blockHeights[i]);
23         }
24         int trappedWater = 0;
25         for (int i = 0; i < n; i++) {
26             int minHeight = Math.min(l_block_Height[i], r_block_Height[i]);
27             trappedWater += Math.max(0, minHeight - blockHeights[i]);
28         }
29         return trappedWater;
30     }
31 }
32 public class Water {
33     public static void main(String[] args) {
34         WaterConservationSystem conservationSystem = new CityBlockConservation();
35         int[] blockHeights = {3,0,0,2,0,4};
36         int trappedWater = conservationSystem.calculateTrappedWater(blockHeights);
37         System.out.println("Trapped Water: " + trappedWater + " units");
38     }
39 }
```

Lab5 > src > JAVA_LABS > Water.java > CityBlockConservation > calculateTrappedWater 13:20 CRLF UTF-8 4 spaces

OUTPUT:



The screenshot displays the IntelliJ IDEA IDE interface. The top toolbar includes icons for file operations, search, and running code. The 'Main.java' tab is active, showing the following Java code:

```
17     for (int i = 1; i < n; i++) {
18         l_block_Height[i] = Math.max(l_block_Height[i - 1], blockHeights[i]);
19     }
20     r_block_Height[n - 1] = blockHeights[n - 1];
21     for (int i = n - 2; i >= 0; i--) {
22         r_block_Height[i] = Math.max(r_block_Height[i + 1], blockHeights[i]);
23     }
24     int trappedWater = 0;
25     for (int i = 0; i < n; i++) {
26         int minHeight = Math.min(l_block_Height[i], r_block_Height[i]);
27         trappedWater += Math.max(0, minHeight - blockHeights[i]);
28     }
```

Below the code editor, the 'Run' window is open, showing the execution output:

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.2.4\lib\idea_rt.jar=6212;"
Trapped Water: 10 units

Process finished with exit code 0
```

The bottom status bar indicates the project path: 'Lab5 > src > JAVA_LAB5 > Water.java > CityBlockConservation > calculateTrappedWater'. The time is 13:20, and the encoding is UTF-8 with 4 spaces.