

Міністерство освіти і науки України
Національний технічний університет України «Київський
політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 6 з дисципліни
«Основи програмування – 2. Методології програмування»

Варіант 23

Виконав студент

ІІ-15, Мочалов Дмитро Юрійович
(шифр, прізвище, ім'я, по батькові)

Перевірів

Муха Ірина Павлівна
(прізвище, ім'я, по батькові)

Київ 2022

Завдання

23. Побудувати два бінарних дерева, елементами якого є цілі числа. Об'єднати їх, уникаючи дублювання елементів в сумарному дереві.

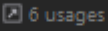
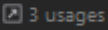
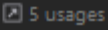
Код C#

```
C# Program.cs × C# Node.cs × C# BinaryTree.cs × C# Functions.cs ×
1 using System;
2 using System.Collections.Generic;
3 using static Lab6.Functions;
4
5 namespace Lab6
6 {
7     internal class Program
8     {
9         public static void Main(string[] args)
10        {
11            Console.Write("Введіть кількість елементів: ");
12            int n = int.Parse(Console.ReadLine());
13            BinaryTree tree1 = new BinaryTree();
14            int elem;
15            for (int i = 0; i < n; i++)
16            {
17                Console.Write("Введіть цілочисельний елемент дерева: ");
18                elem = int.Parse(Console.ReadLine());
19                tree1.Insert(ref tree1.Root, elem);
20            }
21            tree1.print_tree();
22
23            Console.Write("Введіть кількість елементів: ");
24            n = int.Parse(Console.ReadLine());
25            BinaryTree tree2 = new BinaryTree();
26            for (int i = 0; i < n; i++)
27            {
28                Console.Write("Введіть цілочисельний елемент дерева: ");
29                elem = int.Parse(Console.ReadLine());
30                tree2.Insert(ref tree2.Root, elem);
31            }
32            tree2.print_tree();
33            List<int> unitedElement = new List<int>();
34            UnitTree(tree1.Root, unitedElement);
35            UnitTree(tree2.Root, unitedElement);
36            BinaryTree resultTree = new BinaryTree();
37            foreach (var ele: int in unitedElement)
38            {
39                resultTree.Insert(ref resultTree.Root, ele);
40            }
41        }
42    }
43 }
```

```
42         Console.WriteLine("Результующе дерево: ");
43         resultTree.print_tree();
44
45
46     }
47 }
48 }
```

C# Program.cs × C# Node.cs × C# BinaryTree.cs × C# Functions.cs ×

```
1 namespace Lab6
2 {
3     8 usages 3 exposing APIs
4     public class Node
5     {
6         public int Data;
7         public Node LeftNode;
8         public Node RightNode;
9
10        1 usage
11        public Node(int value)
12        {
13            Data = value;
14            LeftNode = null;
15            RightNode = null;
16        }
17    }
18 }
```



```
C# Program.cs × C# Node.cs × C# BinaryTree.cs × C# Functions.cs ×
1  using System;
2
3  namespace Lab6
4  {
5       6 usages
6      public class BinaryTree
7      {
8
9           3 usages
10         public BinaryTree()
11         {
12             Root = null;
13         }
14
15          5 usages
16         public void Insert(ref Node node,int value)
17         {
18             if (node == null)
19             {
20                 node = new Node(value);
21             }
22             else
23             {
24                 if (value < node.Data)
25                 {
26                     Insert(ref node.LeftNode,value);
27                 }
28                 else
29                 {
30                     Insert(ref node.RightNode,value);
31                 }
32             }
33         }
34     }
35 }
```

C# Program.cs × C# Node.cs × C# BinaryTree.cs × C# Functions.cs ×

```

33     private int Levels(Node node)
34     {
35         if (node == null) return 0;
36         return Math.Max(Levels(node.LeftNode), Levels(node.RightNode)) + 1;
37     }
38
39     public void print_tree()
40     {
41         int lvl = Levels(this.Root);
42         for (int i = 0; i < lvl; i++)
43         {
44             Console.Write($"{i}" + "\t");
45         }
46
47         Console.WriteLine();
48         for (int i = 0; i < lvl; i++)
49         {
50             Console.Write("\t");
51         }
52
53         Console.WriteLine("\n");
54         print_tree(this.Root, space: 0);
55     }
56
57     private void print_tree(Node node, int space)
58     {
59         if (node == null) return;
60         print_tree(node.RightNode, ++space);
61         for (int i = 1; i < space; i++)
62         {
63             Console.Write("\t");
64         }
65         Console.WriteLine(node.Data);
66         print_tree(node.LeftNode, space);
67     }
68
69
70

```

```
C# Program.cs × C# Node.cs × C# BinaryTree.cs × C# Functions.cs ×
1      using System.Collections.Generic;
2
3      namespace Lab6
4      {
5           1 usage
6          public static class Functions
7          {
8               4 usages
9              public static void UnitTree(Node node, List<int> result)
10             {
11                 if (node == null)
12                 {
13                     return;
14                 }
15
16                 if (!result.Contains(node.Data))
17                 {
18                     result.Add(node.Data);
19                 }
20
21                 UnitTree(node.LeftNode, result);
22                 UnitTree(node.RightNode, result);
23             }
24         }
25     }
```

Робота програми

C#

C# Program.cs × C# Node.cs × C# BinaryTree.cs × C# Functions.cs ×

Run:    

D:/Repository/Lab6/bin/Debug/Lab6.exe

Введ?ть к?льк?сть элемент?v: 4

Введ?ть ц?лочисельный элемент дерева: 5

Введ?ть ц?лочисельный элемент дерева: 3

Введ?ть ц?лочисельный элемент дерева: 8

Введ?ть ц?лочисельный элемент дерева: 2

```
0      1      2
|      |      |
```

```
      8
```

```
5
```

```
      3
```

```
      2
```

Введ?ть к?льк?сть элемент?v: 5

Введ?ть ц?лочисельный элемент дерева: 9

Введ?ть ц?лочисельный элемент дерева: 1

Введ?ть ц?лочисельный элемент дерева: 6

Введ?ть ц?лочисельный элемент дерева: 10

Введ?ть ц?лочисельный элемент дерева: 8

```
0      1      2      3
|      |      |      |
```

```
      10
```

```
9
```

```
      8
```

```
      6
```

```
      1
```

Результующе дерево:

```
0      1      2      3
|      |      |      |
```

```
      10
```

```
      9
```

```
      8
```

```
      6
```

```
5
```

```
      3
```

```
      2
```

```
      1
```