

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»

«Дерева»

Варіант №4

Виконав студент ІП-14 Берковський Максим Юрійович
(шифр, прізвище, ім'я, по батькові)

Перевірив _____
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота №6

Тема: Дерева

Мета – вивчити особливості організації і обробки дерев.

Хід роботи

Задача.

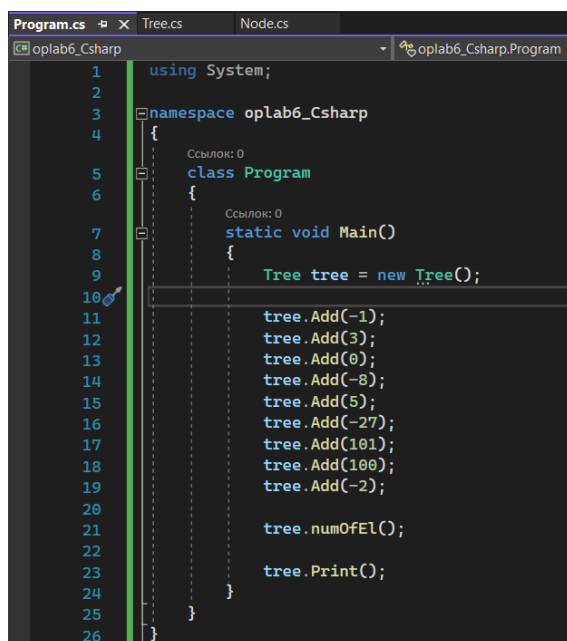
4. Побудувати дерево, елементами якого є числа. Надрукувати дерево. Визначити кількість від'ємних та додатніх елементів дерева.

Постановка задачі.

Для виконання цього завдання спочатку потрібно створити клас, котрий буде шаблоном для створення елемента дерева, який буде вміщати інформацію, покажчик на правий елемент та покажчик на лівий елемент відповідно. Потім створимо клас самого дерева. В ньому потрібно створити метод для додавання елемента в дерево, та метод для виведення дерева в консоль. Також необхідно створити лічильники для підрахунку додатніх та від'ємних елементів в дереві. Збільшувати їх ми будемо одразу після виклику метода додавання елемента. В кінці створимо окремий метод для виведення цих лічильників у консоль.

Випробування коду на C#:

Код:



```
Program.cs | Tree.cs | Node.cs
oplab6_Csharp | oplab6_Csharp.Program
1 using System;
2
3 namespace oplab6_Csharp
4 {
5     class Program
6     {
7         static void Main()
8         {
9             Tree tree = new Tree();
10
11             tree.Add(-1);
12             tree.Add(3);
13             tree.Add(0);
14             tree.Add(-8);
15             tree.Add(5);
16             tree.Add(-27);
17             tree.Add(101);
18             tree.Add(100);
19             tree.Add(-2);
20
21             tree.numOfEL();
22
23             tree.Print();
24         }
25     }
26 }
```

Program.cs

Tree.cs

Node.cs

C# oplab6_Csharp

oplab6_Csharp.Tree

```
1  using System;
2
3  namespace oplab6_Csharp
4  {
5      Ссылка: 3
6      public class Tree
7      {
8          Ссылка: 4
9          private Node root;
10         Ссылка: 3
11         private int count { get; set; }
12         Ссылка: 3
13         private int countPlus { get; set; }
14         Ссылка: 3
15         private int countMinus { get; set; }
16         Ссылка: 3
17         private bool isZero { get; set; }
18         private IComparer<int> _comparer = Comparer<int>.Default;
19
20         Ссылка: 1
21         public Tree()
22         {
23             root = null;
24             count = 0;
25             countPlus = 0;
26             countMinus = 0;
27             isZero = false;
28         }
29
30         Ссылка: 1
31         public void numOfEl()
32         {
33             Console.WriteLine("=====");
34             Console.WriteLine($"Number of elements more than zero: {countPlus}.");
35             Console.WriteLine($"Number of elements less than zero: {countMinus}.");
36             if (isZero) Console.WriteLine("Also there is zero element.");
37             else Console.WriteLine("There is no zero element");
38             Console.WriteLine("=====");
39         }
40
41         Ссылка: 9
42         public bool Add(int Item)
43         {
44             if (Item < 0) countPlus++;
45             else if (Item > 0) countMinus++;
46             else isZero = true;
47
48             if (root == null)
49             {
50                 root = new Node(Item);
51                 count++;
52                 return true;
53             }
54             else
55             {
56                 return Add_Sub(root, Item);
57             }
58         }
59     }
60 }
```

Program.cs

Tree.cs



Node.cs

C# oplab6_Csharp



oplab6_Csharp.Tree

```
48     }
49
50
51     private bool Add_Sub(Node Node, int Item)
52     {
53         if (_comparer.Compare(Node.item, Item) < 0)
54         {
55             if (Node.right == null)
56             {
57                 Node.right = new Node(Item);
58                 count++;
59                 return true;
60             }
61             else
62             {
63                 return Add_Sub(Node.right, Item);
64             }
65         }
66         else if (_comparer.Compare(Node.item, Item) > 0)
67         {
68             if (Node.left == null)
69             {
70                 Node.left = new Node(Item);
71                 count++;
72                 return true;
73             }
74             else
75             {
76                 return Add_Sub(Node.left, Item);
77             }
78         }
79         else
80         {
81             return false;
82         }
83     }
84
85     private void Print()
86     {
87         Print(root, 4);
88     }
89
90     private void Print(Node p, int padding)
91     {
92         if (p != null)
93         {
94             if (p.right != null)
95             {
96                 Print(p.right, padding + 4);
97             }
98             if (padding > 0)
99             {
100                 Console.Write(" ".PadLeft(padding));
101             }
102             if (p.right != null)
103             {
104                 Console.WriteLine("\n");
105                 Console.Write(" ".PadLeft(padding));
106             }
107             Console.WriteLine(p.item.ToString() + "\n ");
108             if (p.left != null)
109             {
110                 Console.WriteLine(" ".PadLeft(padding) + "\\n");
111                 Print(p.left, padding + 4);
112             }
113         }
114     }
115 }
116 }
```

```
Program.cs  Tree.cs  Node.cs  X
oplab6_Csharp  oplab6_Csharp.Node

1  using System;
2
3  namespace oplab6_Csharp
4  {
5      Ссылка: 9
6      public class Node
7      {
8          public int item;
9          public Node right;
10         public Node left;
11
12         Ссылка: 3
13         public Node(int item)
14         {
15             this.item = item;
16         }
17     }
```

Результат:

```
Консоль отладки Microsoft Visual Studio

=====
Number of elements more than zero: 4.
Number of elements less than zero: 4.
Also there is zero element.
=====

      101
       \
        100
       /
      5
     /
    3
   \
   0
  /
 -1
 \
  -2
 /
-8
 \
-27
```

Висновок:

- ознайомився з такою структурою даних, як дерева і навчився розробляти алгоритми та програми з їх застосуванням;
- перевінив роботу програми, випробувавши її у відладчику середовища розробки Visual Studio і отримав задовільний результат.