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»

«Дерева»

Варіант 32

Виконав студент ІП-11 Фукс Вікторія Ігорівна

Перевірив \_\_\_\_\_Вітковська Ірина Павлівна\_\_\_\_\_\_\_\_\_

Київ 2022

**Лабораторна робота 6**

**ДЕРЕВА**

**Мета** – вивчити особливості організації і обробки дерев.

**Постановка задачі**:

****

**Код:**

**C#**

**Program.cs:**

using System;  
  
namespace csharp\_lab6  
{  
 class Program  
 {  
 static void Main(string[] args)  
 {  
 Tree tree = new Tree();  
   
 Console.WriteLine("Enter number of elements in tree: ");  
 int numOfEl = Convert.ToInt32(Console.ReadLine());  
  
 Console.WriteLine("Tree: ");  
 for (int i = 0; i < numOfEl; i++)  
 {  
 tree.Root = tree.AddNode(Rnd(), tree.Root);  
 tree.PrintTree(Console.WindowWidth / 3, 3, tree.Root);  
 }  
   
  
 Console.SetCursorPosition(0, 25);  
 Console.WriteLine($"Average of elements in tree: {(tree.SummaElements(tree.Root))/numOfEl+1}");  
 Console.ReadKey();  
  
 }  
   
 public static int Rnd()  
 {  
 Random rnd = new Random();  
 return rnd.Next(0, 21);  
 }  
 }  
}

**Tree.cs:**

using System;  
  
namespace csharp\_lab6  
{  
 public class Tree  
 {  
 private Node \_root;  
 public Node Root { get => \_root; set => \_root = value; }  
 public Node AddNode(int inputDataNode, Node root)  
 {  
 if (root == null)  
 {  
 root = new Node(inputDataNode);  
 }  
 else  
 {  
 if (inputDataNode < root.Data)  
 {  
 root.Left = AddNode(inputDataNode, root.Left);  
 }  
 else  
 {  
 root.Right = AddNode(inputDataNode, root.Right);  
 }  
 }  
   
 return root;  
 }  
  
 public void PrintTree(int x, int y, Node root, int delta = 0)  
 {  
 if (root != null)  
 {  
 if (delta == 0) delta = x/3;  
 Console.SetCursorPosition(x, y);  
 Console.Write(root.Data);  
 PrintTree(x-delta, y + 2, root.Left, delta / 3);  
 PrintTree(x+delta, y + 2, root.Right,delta / 3);  
 }  
   
 }  
 public int SummaElements(Node root)  
 {  
 if (root == null)  
 return 0;  
 else  
 {  
 int count = 0;  
 count += SummaElements(root.Left);  
 count += SummaElements(root.Right);  
   
 return count + root.Data;  
 }  
 }  
 }  
}

**Node.cs:**

namespace csharp\_lab6  
{  
 public class Node  
 {  
 private int \_data;  
 private Node \_left;  
 private Node \_right;  
   
 public Node()  
 {  
 }  
   
 public Node(int inputDataNode)  
 {  
 Data = inputDataNode;  
 }  
   
 public Node(int data, Node left, Node right)  
 {  
 Data = data;  
 Left = left;  
 Right = right;  
 }  
   
 public int Data { get => \_data; set => \_data = value; }  
 public Node Left { get => \_left; set => \_left = value; }  
 public Node Right { get => \_right; set => \_right = value; }  
 }  
}

**Випробовування алгоритму:**

**C#:**



**Висновок:**

Ми вивчили особливості організації і обробки дерев.

У результаті лабораторної роботи ми написали код на мові програмування С#. Зробили випробування алгоритму.