Министерство образования Республики Беларусь

Министерство науки и высшего образования Российской Федерации

Межгосударственное образовательное учреждение высшего образования

«Белорусско-Российский университет»

Кафедра «Автоматизированные системы управления»

Отчет

по дисциплине «Типы и Структуры Данных»

по лабораторной работе №6

«Представления бинарных деревьев»

Выполнил:

Студент гр. ПИР-221

Сиканве.Б.К

Проверил:

Беккер И. А.

Могилев

2023

**Цель работы:** сформировать знания и умения реализации динамических структур.

Код Программы

namespace BinarySearch

{

public class Node

{

public int Data;

public Node LeftChild;

public Node RightChild;

public void printNode()

{

Console.WriteLine(Data);

}

}

public class BinarySearchTree

{

public Node root;

public BinarySearchTree()

root = null;

}

public void add(int data)//addTotheTree

{

Node newNode = new Node();

newNode.Data = data;

if(root == null)

{

root = newNode;

}

else

{

Node current = root;

Node parent;

while(true)

{

parent = current;

if(data < current.Data)

{

current= current.LeftChild;

if(current ==null)

{

parent.LeftChild= newNode;

break;

}

}

else

{

current = current.RightChild;

if(current == null)

{

parent.RightChild= newNode;

break;

}

}

}

}

}

//How To Traverse or ОБХОД

public void InOrder(Node OurRoot)

{

if (OurRoot!= null)

{

InOrder(OurRoot.LeftChild);

OurRoot.printNode();

InOrder(OurRoot.RightChild);

}

}

public void PrintPreOrder(Node OurRoot)

{

if(OurRoot!= null)

{

OurRoot.printNode();

PrintPreOrder(OurRoot.LeftChild);

PrintPreOrder(OurRoot.RightChild);

}

}

public void PrintPostOrder(Node OurRoot)

{

if (OurRoot != null)

{

PrintPreOrder(OurRoot.LeftChild);

PrintPreOrder(OurRoot.RightChild);

OurRoot.printNode();

}

}

}

internal class Program

{

static void Main(string[] args)

{

BinarySearchTree tree1 = new BinarySearchTree();

tree1.add(50);

tree1.add(40);

tree1.add(60);

tree1.add(20);

tree1.add(45);

tree1.add(55);

tree1.add(65);

Console.WriteLine(tree1);

Console.WriteLine("In-Order Traversal:");

tree1.InOrder(tree1.root);

Console.WriteLine("Pre-Order Traversal:");

tree1.PrintPreOrder(tree1.root);

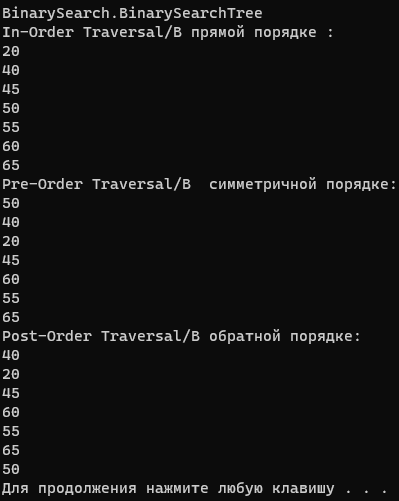
Console.WriteLine("Post-Order Traversal:");

tree1.PrintPostOrder(tree1.root);

}

}

}



Заключение

Если у вас есть несколько экземпляров стека, каждый экземпляр будет иметь свой собственный отдельный набор значений и поддерживать свой собственный порядок элементов. Добавление значения к одному экземпляру стека не влияет на другие экземпляры. Каждый экземпляр стека работает как независимая структура данных last-in, first-out (LIFO)..

If you have multiple instances of a stack, each instance will have its own separate set of values and maintain its own order of elements. Adding a value to one instance of the stack does not affect the other instances. Each stack instance operates as an independent last-in, first-out (LIFO) data structure.