Assignment - 9

Write a program to implement all the binary tree traversal methods using classes.

Rudra Chinhara
202117
B. Tech. CSE, 2nd Year
Central University of Haryana

https://github.com/RPChinhara-202117/DSA_Assignments.git

Code:

75 lines (56 sloc) | 1.36 KB

Raw Blame

```
#include <iostream>
using namespace std;

class Node
{
public:
    int data;
    Node *left, *right;

    Node(int data)
    {
        this->data = data;
        left = right = NULL;
    }

    // Printing a Binary Tree according to PostOrder Traversal

    void PostOrder(Node *node)
    {
        if(node == NULL)
            return;

        PostOrder(node->left);
        PostOrder(node->right);
        cout<<node->data<<" ";
    }

    // Printing a Binary Tree according to InOrder Traversal

    void InOrder(Node *node)
    {
        if(node == NULL)
            return;
```

```
    void InOrder(Node *node)
    {
        if(node == NULL)
            return;

        InOrder(node->left);
        cout<<node->data<<" ";
        InOrder(node->right);
    }

    // Printing a Binary Tree according to PreOrder Traversal

    void PreOrder(Node *node)
    {
        if(node == NULL)
            return;

        cout<<node->data<<" ";
        PreOrder(node->left);
        PreOrder(node->right);
    }
};



int main()
{
    Node *root = new Node(1);
    root->left = new Node(2);
    root->right = new Node(3);
    root->left->left = new Node(4);
    root->left->right = new Node(5);

    cout<<"\nPreOrder Traversal of Binary Tree is \n";
    (*root).PreOrder(root);
```

```
        PreOrder(node->left);
        PreOrder(node->right);
    }
};



int main()
{
    Node *root = new Node(1);
    root->left = new Node(2);
    root->right = new Node(3);
    root->left->left = new Node(4);
    root->left->right = new Node(5);

    cout<<"\nPreOrder Traversal of Binary Tree is \n";
    (*root).PreOrder(root);

    cout<<"\nInOrder Traversal of Binary Tree is \n";
    (*root).InOrder(root);

    cout<<"\nPostOrder Traversal of Binary Tree is \n";
    (*root).PostOrder(root);

    cout<<endl;

    return 0;
}
```

Terminal Output: