

Binary Tree - Homework

AKHMEDOV YUSUFJON MAKHAMMAD UGLI

ID: 202438404 Computer Engineer Student

CODE using Python

Explanation

data-structure > Trees > Homework >  main.py > ...

```
1  class Node:
2      def __init__(self, value):
3          self.value = value
4          self.left = None
5          self.right = None
6
7  class BinaryTree:
8      def __init__(self):
9          self.root = None
10
11     def add_node(self, value):
12         if self.root is None:
13             self.root = Node(value)
14         else:
15             self._insert(self.root, value)
16
17     def _insert(self, current, value):
18         if value < current.value:
19             if current.left is None:
20                 current.left = Node(value)
21             else:
22                 self._insert(current.left, value)
23         else:
24             if current.right is None:
25                 current.right = Node(value)
26             else:
27                 self._insert(current.right, value)
28
29     def preorder(self, node):
30         if node is None:
31             return []
32         return [node.value] + \
33             self.preorder(node.left) + \
34             self.preorder(node.right)
35
```

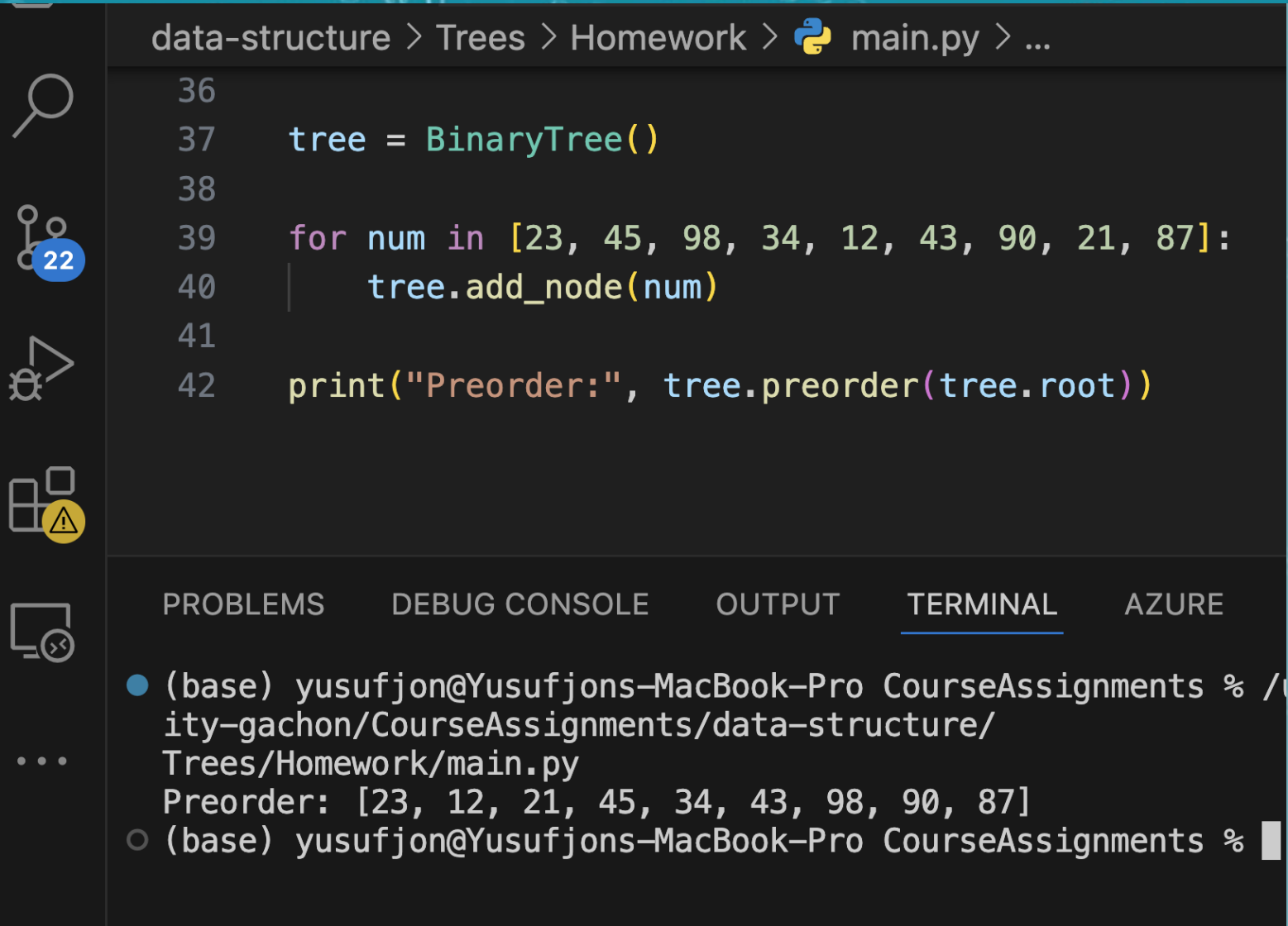
We have two classes named **BinaryTree** and **Node**.

BinaryTree has `add_node` and `preorder` methods which can be used to make a tree using nodes and to show the nodes' values in **preorder Traversal**. (Other traversal methods can be added to this class)

Node has **value** which is a number, then left and right properties which store node in the same type but having different values, for example, **left** and **right** nodes we can say.

[GitHub Link](#)

Code Output



The image shows a VS Code editor window with a Python file named `main.py` in the `data-structure > Trees > Homework` directory. The code defines a `BinaryTree` class and adds nodes to it. The terminal output shows the pre-order traversal of the tree.

```
data-structure > Trees > Homework > main.py > ...  
36  
37     tree = BinaryTree()  
38  
39     for num in [23, 45, 98, 34, 12, 43, 90, 21, 87]:  
40         tree.add_node(num)  
41  
42     print("Preorder:", tree.preorder(tree.root))
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL AZURE

- (base) yusufjon@Yusufjons-MacBook-Pro CourseAssignments % /
ity-gachon/CourseAssignments/data-structure/
Trees/Homework/main.py
Preorder: [23, 12, 21, 45, 34, 43, 98, 90, 87]
- (base) yusufjon@Yusufjons-MacBook-Pro CourseAssignments %