Pseudo code for binary tree assignment

# Treenode ()

Fields

ID : this is the special number for each item that is completely unique to the item no item can have the same id int .

Name: The name of the item String

Price: The value of the item in terms of currency double.

Amount: This is the total number of items with the same specific id int.

Treenode()

The treenode methods are meant to initialise the fields of the Treenodes and set them to a standard value that they would always start with.

# Binary Tree()

Fields

Treenode root: This is the very first treenode in the binary tree

Treenode Previous: This is the treenode that is always one behind the current treenode.

Treenode Current: This is the treenode that the program is constantly checking.

BinaryTree(): Constructor method that sets the root to null

addTree(): This method checks if the root is null if it is then sets the first values that come into it as the root, its input parameters are name id and price and if the root isn’t null then it passes the information into another add method.

addTree2(): This method does the same thing as the other addTree method but addTree2 adds thing to the binary tree that is going to be written to a file.

Addnode(): The add node methode checks if the id of the input is greater than or less than the root if it is less it keeps going too the left of the root while checking if the current tree node is null. If it isn’t then it sets that tree node as the new values that come in. if it is it uses recursion and just goes further down the Binary tree.

Addnode2():Addnode 22 does the same thing as addnode but for the file writer.

Deleteroot(): The delete root method checks if the node that wants to be deleted is the root, if it isn’t then it passes through another method, but if it is it checks for the right most node on the left side. Deletes the root node and replaces the root node with the right most node on the left side.

Rootdeltercheckleft(): This method handles the main part of delting the root for the treenode this method checks for the right most node on the left side of the root using recursion and when it finds it replaces the root with that node.

Deletenode(): This method handles the delete for the rest of the tree that is not the root, this method checks if the node that is currently going to be deleted is attached to anything, this method handles nodes that are not attacked to anything or attached to one thing.

Deletew2left1() && Deletew2left2: This method is used to delete tree nodes that are attached to 2 other nodes on the left side of the tree.

Deletew2lright1() && Deletew2right2: This method is used to delete tree nodes that are attached to 2 other nodes on the right side of the tree.

searchRoot():This method checks if the node that is being searched for is the root, if it is not it will pass the user input to the searchnode method.

Searchnode(): This method searches for the node based on the user input entered

PostOrderPrint(): This method prints the tree in the postOrder.

inOrderPrint(): This method prints the tree in the postOrder.

PreOrderPrint(): This method prints the tree in the postOrder.

PrintTotalPrice(): This method prints the total price of all the treenodes in the binary tree.