

Oefening: Boomstructuur opbouwen en recursief doorlopen

Boomstructuren zijn een klassiek én zeer nuttig voorbeeld van recursie. Hieronder geef ik je een volledige oefening waarin we:

1. Een **boomstructuur samenstellen** met knopen.
2. Deze boom **recursief doorlopen** (pre-order).
3. Eventueel uitbreidbaar maken met dieptes, zoekopdrachten, enz.

We maken een boom waarin elke knoop een naam heeft, en 0 of meer kinderen kan bevatten (typisch voor een **N-ary tree**).

Stap 1: Boomknoopklasse (`TreeNode`)

```
using System;
using System.Collections.Generic;

class TreeNode
{
    public string Naam { get; set; }
    public List<TreeNode> Kinderen { get; set; }

    public TreeNode(string naam)
    {
        Naam = naam;
        Kinderen = new List<TreeNode>();
    }

    public void VoegKindToe(TreeNode kind)
    {
        Kinderen.Add(kind);
    }
}
```

Stap 2: Recursieve doorloop (pre-order traversie)

```
static void PrintBoom(TreeNode node, int diepte = 0)
{
    // Inspringing volgens diepte
    Console.WriteLine(new string('-', diepte * 2) + node.Naam);

    // Recursief alle kinderen aflopen
    foreach (var kind in node.Kinderen)
    {
        PrintBoom(kind, diepte + 1);
    }
}
```

Stap 3: Voorbeeldboom samenstellen en aanroepen

```
class Program
{
    static void Main(string[] args)
    {
        // Boomstructuur opbouwen
        TreeNode root = new TreeNode("CEO");

        TreeNode cto = new TreeNode("CTO");
        cto.VoegKindToe(new TreeNode("Dev Manager"));
        cto.VoegKindToe(new TreeNode("QA Manager"));

        TreeNode cfo = new TreeNode("CFO");
        TreeNode accountant = new TreeNode("Accountant");
        accountant.VoegKindToe(new TreeNode("Boekhouder Junior"));
        cfo.VoegKindToe(accountant);

        root.VoegKindToe(cto);
        root.VoegKindToe(cfo);
        root.VoegKindToe(new TreeNode("COO"));

        // Recursief printen
        PrintBoom(root);

        Console.WriteLine("Druk op een toets om af te sluiten...");
        Console.ReadKey();
    }
}
```

Output

```
CEO
--CTO
----Dev Manager
----QA Manager
--CFO
----Accountant
-----Boekhouder Junior
--COO
```

Uitleg

- Elke knoop (TreeNode) kan meerdere kinderen hebben.
 - De PrintBoom-methode **doorloopt recursief** de boom, en drukt de namen af met visuele inspringing.
 - We gebruiken *diepte* om het niveau bij te houden, handig voor vormgeving of logica (bv. laden op basis van diepte).
-