

Forum:[Hack Forums](#)

Thread: [A small recursive utility.](#)

Ethereal - October 2, 2011, 12:53 am

Earlier, I lost track of the total number of posts I had in TAFWA. For those who don't know, that's just TA's way of keeping track of helpful posts. They are stored in this format:

Code (Amount) Month Day, Year

I wanted to add up all of the amounts, so I wrote a quick little recursive algorithm, for a little practice, I guess.

By the way, I know this isn't a beautiful algorithm, but it works and that is what matters; its an example to help people understand basic recursion.

PHP Code

```
<?php
private int GetTotalNumber(string text)
{
    int count = 0;
    foreach (string line in text.Split('\n'))
    {
        count += GetNumber(line.Trim());
    }
    return count;
}

private int GetNumber(string line, int counter = 0, int temp = 0)
{
    if (line == "") return temp;
    else if (line[counter] == '(') return GetNumber(line, counter + 1, temp);
    else if (line[counter] == ')') return temp;
    else if (char.IsNumber(line[counter]))
    if (char.IsNumber(line[counter + 1]))
    {
        string tempStr = line.Substring(counter, counter + 1);
        return GetNumber(line, counter + 2, temp + Convert.ToInt32(tempStr));
    }
    else
        return GetNumber(line, counter + 1, temp + (((char)(Convert.ToInt32(line[counter]))) - '0'));
    else
        return temp;
}
```

Input:

Code (15)

(3)

(8)

(7)

Output:

Code 33

Works for anything up to:

Code (99)

which is enough for its use.

So yeah, just a basic example. Hadn't released anything public for some time, but decided I might as well post this.

ubbelol - October 2, 2011, 5:23 am

That is some good stuff Ethereum, recursive methods is something you have to work with whether you like it or not when you get further in your programming.

Here's another example I made to recursively loop through a treeviews nodes to get them all (needed this for my obfuscator project). It's kind of stupid that treeview doesn't have a property implemented to get all nodes. :|

```
Code private List<TreeNode> RetrieveAllNodes(TreeView treeView)
```

```
{
    List<TreeNode> outList = new List<TreeNode>();

    foreach (TreeNode node in treeView1.Nodes)
        GetSubNodes(node, outList);

    return outList;
}
```

```
private void GetSubNodes(TreeNode node, List<TreeNode> outList)
```

```
{
    outList.Add(node);

    foreach (TreeNode _node in node.Nodes)
        GetSubNodes(_node, outList);
}
```

Probably some better way to do this?

Nonmod - October 2, 2011, 5:50 am

UbbeLoL, magic answer is binary trees. Traversal alone saved up my cpu time by 40%+. Use a composite design pattern.

Both combined together, produce magic of infinite amounts.

KingCobraXD - October 2, 2011, 6:59 am

Nice share. I love recursion although it can eat up a lot of memory.

Ethereal - October 2, 2011, 8:51 am

(Oct 2, 2011 05:23 AM)UbbeLoL Wrote: [\[CITE\]](#) That is some good stuff Ethereum, recursive methods is something you have to work with whether you like it or not when you get further in your programming.

Here's another example I made to recursively loop through a treeviews nodes to get them all (needed this for my obfuscator project). It's kind of stupid that treeview doesn't have a property implemented to get all nodes. :|

```
Code private List<TreeNode> RetrieveAllNodes(TreeView treeView)
{
    List<TreeNode> outList = new List<TreeNode>();

    foreach (TreeNode node in treeView1.Nodes)
        GetSubNodes(node, outList);

    return outList;
}

private void GetSubNodes(TreeNode node, List<TreeNode> outList)
{
    outList.Add(node);

    foreach (TreeNode _node in node.Nodes)
        GetSubNodes(_node, outList);
}
```

Probably some better way to do this?

(Oct 2, 2011 05:50 AM)Nonmod Wrote: [\[CITE\]](#) UbbeLol, magic answer is binary trees. Traversal alone saved up my cpu time by 40%+.

Use a composite design pattern.

Both combined together, produce magic of infinite amounts.

This.

(Oct 2, 2011 06:59 AM)KingCobraXD Wrote: [\[CITE\]](#) Nice share. I love recursion although it can eat up a lot of memory.

Yeah, that's why you have to be careful you don't forkbomb yourself.