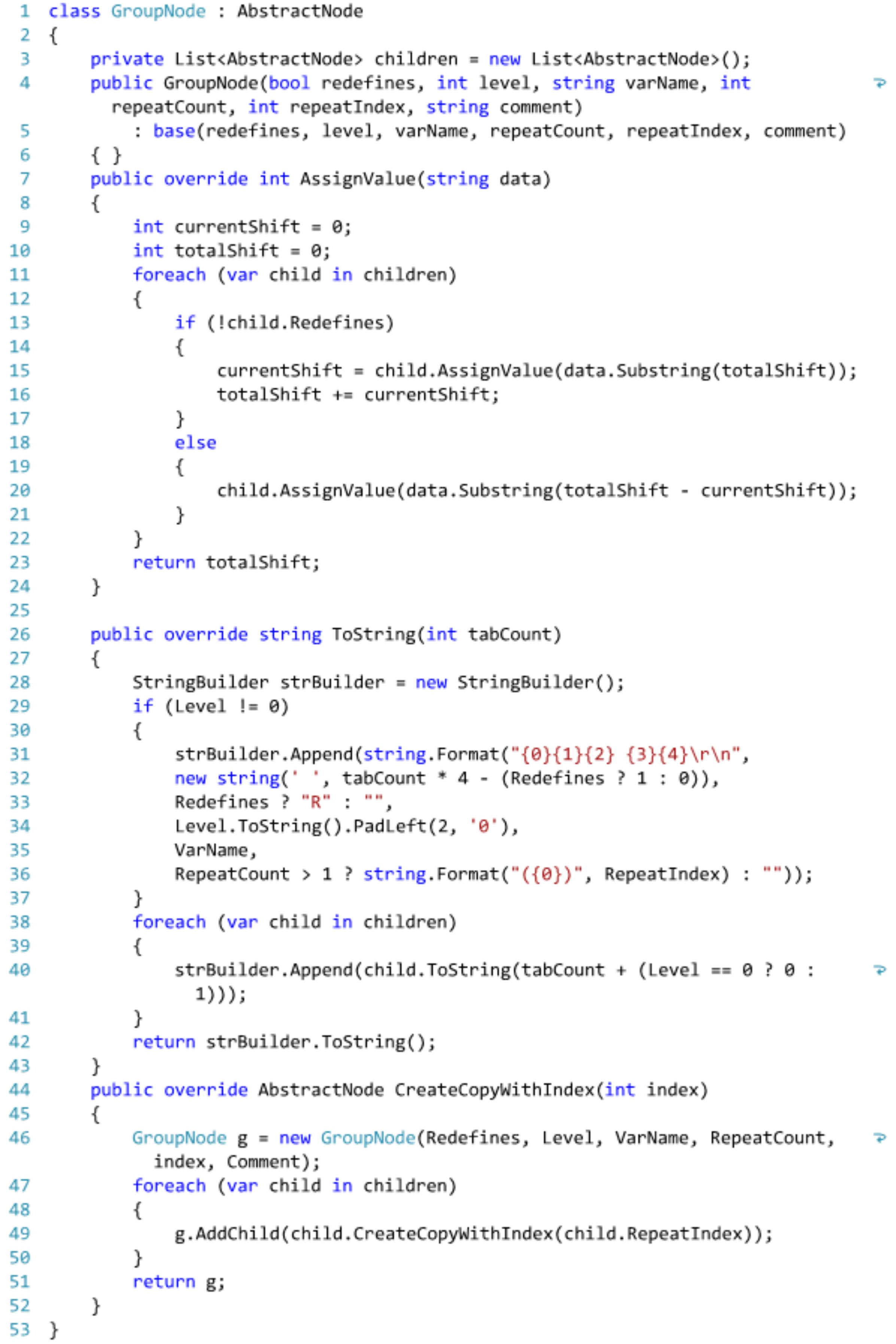

```csharp
class GroupNode : AbstractNode
{
    private List<AbstractNode> children = new List<AbstractNode>();
    public GroupNode(bool redefines, int level, string varName, int repeatCount, int repeatIndex, string comment)
        : base(redefines, level, varName, repeatCount, repeatIndex, comment)
    { }
    public override int AssignValue(string data)
    {
        int currentShift = 0;
        int totalShift = 0;
        foreach (var child in children)
        {
            if (!child.Redefines)
            {
                currentShift = child.AssignValue(data.Substring(totalShift));
                totalShift += currentShift;
            }
            else
            {
                child.AssignValue(data.Substring(totalShift - currentShift));
            }
        }
        return totalShift;
    }

    public override string ToString(int tabCount)
    {
        StringBuilder strBuilder = new StringBuilder();
        if (Level != 0)
        {
            strBuilder.Append(string.Format("{0}{1}{2} {3}{4}\r\n",
            new string(' ', tabCount * 4 - (Redefines ? 1 : 0)),
            Redefines ? "R" : "",
            Level.ToString().PadLeft(2, '0'),
            VarName,
            RepeatCount > 1 ? string.Format("({0})", RepeatIndex) : ""));
        }
        foreach (var child in children)
        {
            strBuilder.Append(child.ToString(tabCount + (Level == 0 ? 0 : 1)));
        }
        return strBuilder.ToString();
    }
    public override AbstractNode CreateCopyWithIndex(int index)
    {
        GroupNode g = new GroupNode(Redefines, Level, VarName, RepeatCount, index, Comment);
        foreach (var child in children)
        {
            g.AddChild(child.CreateCopyWithIndex(child.RepeatIndex));
        }
        return g;
    }
}
```