

```

1 class ValueNode : AbstractNode
2 {
3     public string Type { get; private set; }
4     public int Length { get; private set; }
5     public string Value { get; private set; }
6
7     public ValueNode(bool redefines, int level, string varName, string type,
8         int length, int repeatCount, int repeatIndex, string comment)
9         : base(redefines, level, varName, repeatCount, repeatIndex, comment)
10    {
11        Type = type;
12        Length = length;
13    }
14
15    public override AbstractNode CreateCopyWithIndex(int index)
16    {
17        return new ValueNode(Redefines, Level, VarName, Type, Length,
18            RepeatCount, index, Comment);
19    }
20
21    public override int AssignValue(string data)
22    {
23        Value = data.Substring(0, Length);
24        return Length;
25    }
26
27    public override string ToString(int tabCount)
28    {
29        if (Level == 0)
30        {
31            return "";
32        }
33        return string.Format("{0}{1}{2} {3}{4}={5}\r\n",
34            new string(' ', tabCount * 4 - (Redefines ? 1 : 0)),
35            Redefines ? "R" : "",
36            Level.ToString().PadLeft(2, '0'),
37            VarName,
38            (RepeatCount > 1) ? "(" + RepeatIndex + ")" : "",
39            Value);
40    }
41
42    public override void AddChild(AbstractNode child)
43    {
44        throw new InvalidOperationException(this.VarName + ": " + "Werteknoten
45            können keine Kindknoten haben!");
46    }
47 }

```