

# Sortieralgorithmen: MergeSort

## Quellcode

```
1. Module Module1
2. 'liste mit 15 Elementen, welches später sortiert werden soll deklarieren
4. Dim list_to_be_sorted(15) As Integer
6. 'Hilfsroutine um eine Liste von Integern in der Konsole auszugeben
7. Sub print_list(ByRef liste As List(Of Integer))
8. For Each element In liste
9. Console.Write(element.ToString & " ")
10. Next
11. Console.WriteLine("-")
12. End Sub
13. 'Methode um ein liste mit Zufallswerten zu initialisieren
15. Sub initialize_list(ByRef liste() As Integer)
16. 'Zufallsgenerator initialisieren
17. Dim r As New Random(System.DateTime.Now.Millisecond)
18. 'Jedem Element der Schleife einen zufälligen Wert zuweisen
19. For i As Integer = 0 To liste.Length - 1
20. 'Zufallszahl zwischen 0 und 10.000 erzeugen und zuweisen
21. liste(i) = r.Next(0, 10000)
22. Next
23. End Sub
24. ' MergeSort implementation
26. Sub MergeSort(ByRef szArray As List(Of Integer), ByVal nLower As Integer, ByVal nUpper As Integer)
27. ' Check for array size
29. Dim szSwap As String
30. If (nUpper - nLower) = 1 Then
32. ' Swap strings if necessary
33. If szArray(nLower).CompareTo(szArray(nUpper)) > 0 Then
34. szSwap = szArray(nLower)
35. szArray(nLower) = szArray(nUpper)
36. szArray(nUpper) = szSwap
37. End If
38. ElseIf (nUpper - nLower) > 1 Then
40. ' Sort each half and merge
42. MergeSort(szArray, nLower, (nLower + nUpper) / 2)
43. MergeSort(szArray, (nLower + nUpper) / 2 + 1, nUpper)
44. Merge(szArray, nLower, (nLower + nUpper) / 2 + 1, nUpper)
46. End If
47. End Sub
49. Sub Merge(ByRef szArray As List(Of Integer), ByVal nLower As Integer, ByVal nMiddle As Integer, ByVal nUpper
    As Integer)
52. Dim nIndex As Integer
53. Dim szBuffer As New List(Of Integer)()
54. Dim i, j As Integer
56. i = nLower
57. j = nMiddle
58. While i < nMiddle And j <= nUpper
59. If (szArray(i).CompareTo(szArray(j)) < 0) Then
60. szBuffer.Add(szArray(i))
61. i = i + 1
62. Else
63. szBuffer.Add(szArray(j))
64. j = j + 1
65. End If
```

```

68. End While
69. While i < nMiddle
70. szBuffer.Add(szArray(i))
71. i = i + 1
72. End While
73. While j <= nUpper
74. szBuffer.Add(szArray(j))
75. j = j + 1
76. End While
78. nIndex = 0
79. For i = nLower To nUpper
80. szArray(i) = szBuffer(nIndex)
81. nIndex = nIndex + 1
82. Next i
83. End Sub
86. Sub Main()
87. Dim al As List(Of Integer) = New List(Of Integer)
88. initialize_list(list_to_be_sorted)
89. al.AddRange(list_to_be_sorted)
90. print_list(list_to_be_sorted)
92. MergeSort(al, 0, list_to_be_sorted.Length - 1)
93. print_list(al)
94. Console.ReadKey()
95. End Sub
96. End Module

```

Alles anzeigen

Ich werde hier nicht auf die Spezifika des Algorithmus eingehen, da das sprachenunabhängig wäre und eher in die Rubrik Allgemein gehört. Es sei an dieser Stelle auf den Wikipedia-Artikel verwiesen, der eigentlich alles Nötige erklärt.