

# Informatik 2 mit BlueJ

Ein Kurs für die Stufe 12

von Ulrich Helmich

© Mai 2009  
zuletzt überarbeitet: 30. März 2010

# Inhalt

<b>Folge 11 - Vokabelliste</b>	<b>6</b>
<b>Folge 12 - Suchalgorithmen</b>	<b>7</b>
<b>12.1 Lineare und binäre Suche</b>	<b>7</b>
<i>12.1.1 Binäre Suche</i>	7
<i>12.1.2 Lineare Suche</i>	7
<i>12.1.3 Vergleich der beiden Suchverfahren</i>	7
<b>12.2 Lineare Suche</b>	<b>8</b>
<i>12.2.1 Analyseverfahren</i>	8
<i>12.2.2 Ergebnisse</i>	8
<i>12.2.3 Optimierungen der linearen Suche</i>	10
<b>12.3 Binäre Suche</b>	<b>11</b>
<i>12.3.1 Analyseverfahren</i>	11
<i>12.3.2 Ergebnisse</i>	12
<i>12.3.3 Genauere Analyse</i>	12
<i>12.3.4 Algorithmus der binären Suche</i>	13
<b>Folge 13 - Quicksort</b>	<b>15</b>
<b>13.1 Grundprinzip des Quicksort</b>	<b>15</b>
<b>13.2 Der Quicksort im Detail</b>	<b>16</b>
<b>13.3 Optimierungen des Quicksort</b>	<b>21</b>

<b>Folge 14 - Abstrakte Datentypen</b>	<b>23</b>
14.1 Datentypen und Datenstrukturen	23
14.2 Abstrakte Datentypen	24
14.3 Arbeitsweise eines Stacks	25
14.4 Implementierung einer Stack-Klasse	26
14.5 Die Klasse Stack im Zentralabitur NRW	31
14.6 Aufgaben mit der Klasse Stack	34
14.6.1 Aufgabe: Kopieren eines Stacks	34
14.6.2 Aufgabe: Zusammenfügen von zwei Stacks	39
14.7 Der ADT Queue	40
14.8 Der ADT Dictionary	42
14.9 Abstrakte Datentypen	43
14.10 Der ADT Stack - eine axiomatische Sicht	45
<b>Folge 15/16</b>	<b>47</b>
<b>Folge 17 - Zeiger</b>	<b>48</b>
17.1 Was sind Zeiger?	48
17.2 Zeiger und Referenzen	49
17.3 Ein dynamischer Stack	50
17.4 Eine dynamische sortierte Liste	62

<b>17.5 Implementation der Liste</b>	<b>76</b>
<i>17.5.1 Zielsetzung</i>	<i>76</i>
<i>17.5.2 Erweiterung der Klasse List zur NRW-List</i>	<i>77</i>
<b>17.6 Expertenaufgaben</b>	<b>88</b>
<i>17.6.1 Aufgabe: Vokabelliste</i>	<i>88</i>
<i>17.6.2 Aufgabe: Ein Super-Lexikon</i>	<i>91</i>
<b>Folge 18 - Vererbung</b>	<b>93</b>
<b>18.1 Ein einfacher Fall der Vererbung</b>	<b>93</b>
<i>Exkurs: Vererbung in Java</i>	<i>102</i>
<b>18.2 Heterogene Listen</b>	<b>107</b>
<b>18.3 Eine graphische heterogene Liste</b>	<b>118</b>
<b>Folge 19 - Bäume</b>	<b>120</b>
<b>19.1 Binärbäume - Allgemeines</b>	<b>120</b>
<b>19.2 Binäre Suchbäume</b>	<b>123</b>
<i>19.2.1 Grundlegendes, Rekursivität</i>	<i>123</i>
<i>19.2.2 Vorteile binärer Suchbäume</i>	<i>124</i>
<b>19.3 Binäre Suchbäume mit BlueJ</b>	<b>127</b>
<b>19.4 Eine rekursive Methode zum Anzeigen der Elemente</b>	<b>130</b>
<i>19.4.1 Allgemeines</i>	<i>130</i>
<i>19.4.2 Arbeitsweise von showR()</i>	<i>131</i>
<b>19.5 Eine insert()-Methode</b>	<b>134</b>
<i>Fall 1 - Der Baum ist leer</i>	<i>134</i>

<b>Fall 2 - Der Baum enthält genau ein Element</b>	<b>134</b>
<b>Fall 3 - Der Baum enthält viele Elemente</b>	<b>134</b>
<b>19.6 Eine rekursive insert-Methode</b>	<b>137</b>
<b>Fall 1 - Der Baum ist leer</b>	<b>137</b>
<b>Fall 2 - Der Baum ist nicht leer</b>	<b>137</b>
<b>Quelltext</b>	<b>137</b>
<b>Quelltext-Analyse</b>	<b>138</b>
<b>19.7 Der ADT „binärer Suchbaum“</b>	<b>140</b>
<b>Implementierung mit Hilfe eines Arrays</b>	<b>141</b>
<b>19.8 Die Klasse Ordered Tree</b>	<b>143</b>
<b>19.9 Das Löschen von Elementen</b>	<b>144</b>
<b>19.9.1 Suchen des zu löschenden Elementes</b>	<b>144</b>
<b>19.9.2 Das Löschen - Fall 1</b>	<b>145</b>
<b>19.9.3 Das Löschen - Fall 2</b>	<b>147</b>
<b>19.9.4 Das Löschen - Fall 3</b>	<b>149</b>
<b>19.10 Ausgeglichene Bäume</b>	<b>151</b>
<b>19.10.1 Problemstellung</b>	<b>151</b>
<b>19.10.2 Einfügen in einen ausgeglichenen Baum</b>	<b>153</b>
<b>19.10.3 Vorteile von ausgeglichenen Bäumen</b>	<b>155</b>
<b>19.11 "B-Bäume"</b>	<b>157</b>
<b>19.11.1 Grundidee</b>	<b>157</b>
<b>19.11.2 Eine mögliche Implementation in Java</b>	<b>161</b>
<b>19.11.3 "Richtige" B-Bäume</b>	<b>166</b>