

# Informatikabitur – Softwareentwicklung – Algorithmische Grundstrukturen

## Ergänzung zum Tafelwerk

Name	Verbale Formulierung	Struktogramm	Java-Implementation												
<b>Anweisung</b>															
Anweisung	Anweisung	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">Anweisung</div>	Anweisung;												
<b>Sequenz/Verbundanweisung/Block</b>															
Sequenz	Anweisung 1 Anweisung 2 ... Anweisung N	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">Anweisung 1</div> <hr style="width: 100%;"/> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">Anweisung 2</div> <hr style="width: 100%;"/> <div style="text-align: center;">...</div> <hr style="width: 100%;"/> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">Anweisung N</div>	<pre>{   Anweisung_1;   Anweisung_2;   ...   Anweisung_N; }</pre>												
<b>Auswahl/Selektion/Alternative</b>															
Einseitige Auswahl	WENN Bedingung DANN Sequenz	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <table style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center; border-bottom: none;">Bedingung</td> </tr> <tr> <td style="width: 50%; text-align: center; border-right: none;">wahr</td> <td style="width: 50%; text-align: center; border-left: none;">falsch</td> </tr> <tr> <td colspan="2" style="border-top: none;">Sequenz</td> </tr> </table> </div>	Bedingung		wahr	falsch	Sequenz		<pre>if (Bedingung)   Sequenz</pre>						
Bedingung															
wahr	falsch														
Sequenz															
Zweiseitige Auswahl	WENN Bedingung DANN Sequenz 1 SONST Sequenz 2	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <table style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center; border-bottom: none;">Bedingung</td> </tr> <tr> <td style="width: 50%; text-align: center; border-right: none;">wahr</td> <td style="width: 50%; text-align: center; border-left: none;">falsch</td> </tr> <tr> <td style="border-right: none;">Sequenz 1</td> <td style="border-left: none;">Sequenz 2</td> </tr> </table> </div>	Bedingung		wahr	falsch	Sequenz 1	Sequenz 2	<pre>if (Bedingung)   Sequenz_1 else   Sequenz_2</pre>						
Bedingung															
wahr	falsch														
Sequenz 1	Sequenz 2														
Mehrseitige Auswahl	FALLS Selektor FALL 1: Sequenz 1 FALL 2: Sequenz 2 ... SONST: Sequenz N	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <table style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="4" style="text-align: center; border-bottom: none;">Selektor</td> </tr> <tr> <td style="width: 25%; text-align: center; border-right: none;">1</td> <td style="width: 25%; text-align: center; border-right: none;">2</td> <td style="width: 25%; text-align: center; border-right: none;">...</td> <td style="width: 25%; text-align: center;">sonst</td> </tr> <tr> <td style="border-right: none;">Seq 1</td> <td style="border-right: none;">Seq 2</td> <td style="border-right: none;">...</td> <td>Seq N</td> </tr> </table> </div>	Selektor				1	2	...	sonst	Seq 1	Seq 2	...	Seq N	<pre>switch (Selektor) {   case 1:     Seq_1     break;   case 2:     Seq_2;     break;   ...   default:     Seq_n;     break; }</pre>
Selektor															
1	2	...	sonst												
Seq 1	Seq 2	...	Seq N												

# Informatikabitur – Softwareentwicklung – Algorithmische Grundstrukturen

## Ergänzung zum Tafelwerk

Name	Verbale Formulierung	Struktogramm	Java-Implementation
<b>Schleife/Iteration/Repetition/Wiederholung</b>			
Kopfgesteuerte Schleife	SOLANGE Bedingung FÜHRE AUS Sequenz	<pre> graph TD     A[SOLANGE Bedingung] --&gt; B[Sequenz]         </pre>	<b>while</b> (Bedingung) Sequenz
Fußgesteuerte Schleife	WIEDERHOLE Anweisungsfolge SOLANGE Sequenz	<pre> graph TD     A[Sequenz] --&gt; B[SOLANGE Bedingung]         </pre>	<b>do</b> Sequenz <b>while</b> (Bedingung);
Wiederholung mit fester Anzahl	WIEDERHOLE n MAL Sequenz	<pre> graph TD     A[WIEDERHOLE n MAL] --&gt; B[Sequenz]         </pre>	<b>for</b> (int i = 1; i <= n; i ++) Sequenz
Zählschleife aufwärts	FÜR i = a BIS e SCHRITTWEITE s FÜHRE AUS Sequenz	<pre> graph TD     A[FÜR i = a BIS e SCHRITT s] --&gt; B[Sequenz]         </pre>	<b>for</b> (int i = a; i <= e; i += s) Sequenz
Zählschleife abwärts	FÜR i = a HINAB ZU e SCHRITTWEITE s FÜHRE AUS Sequenz	<pre> graph TD     A[FÜR i = a BIS e SCHRITT s] --&gt; B[Sequenz]         </pre>	<b>for</b> (int i = a; i >= e; i -= s) Sequenz
Listendurchlauf	FÜR JEDES element DER liste VOM typ FÜHRE AUS Sequenz	<pre> graph TD     A[FÜR liste] --&gt; B[Sequenz]         </pre>	<b>for</b> (typ element: liste) Sequenz