

Java Befehlsübersicht LS2

- Übersicht über die benötigten Java-Befehle und Strukturen, so wie sie in LS2 benötigt werden
- Teilweise vereinfachende Darstellung
- Nicht notwendigerweise vollständig
- Ohne Gewähr

Aufgabe	Schlüsselwort / Code allgemein	Beispiel
1. Kontrollstrukturen		
Einseitiges if-Statement	<pre>if(Bedingung){ ... }</pre>	<pre>if(i < 99) { System.out.println("in Ordnung"); }</pre>
Zweiseitiges if-Statement	<pre>if(Bedingung){ ... } else { ... }</pre>	<pre>if(i < 99) { System.out.println("in Ordnung"); } else { System.out.println("zu viel"); }</pre>
swich-Statement	<pre>switch(Variable){ case 1: ... break; case 2: ... break; default: ... }</pre>	<pre>switch(i) { case 1: System.out.println("i ist eins"); break; case 2: System.out.println("i ist zwei"); break; case 3: System.out.println("i ist drei"); break; default: System.out.println("i größer drei"); }</pre>
for-Schleife	<pre>for(Start; Bedingung; Änderung){ ... }</pre>	<pre>for(i = 1; i < 20; i++) { System.out.println(i); }</pre>
while-Schleife	<pre>while(Bedingung) { ... }</pre>	<pre>i = 1; while(i < 20) { System.out.println(i); i++; }</pre>
do-while-Schleife	<pre>do { ... } while(Bedingung);</pre>	<pre>i = 1; do { System.out.println(i); i++; } while(i < 20);</pre>
2. Bedingungen		
Vergleichsoperatoren	<, <=, >, >=, ==, !=	x <= y, z == true
Logische Verknüpfungen	&&, , ! (UND, ODER, NICHT)	(x < y) && (z != false)
3. Schleifensprünge		
Mit nächstem Schleifendurchlauf weitermachen	continue;	
Schleife ganz verlassen	break;	

Aufgabe	Schlüsselwort / Code allgemein	Beispiel
4. Datei-Eingabe und Datei-Ausgabe		
Ausgabe (mit der Klasse FileWriter)	<pre>import java.io.*; public class Beispiel_08 { public static void main(String[] args) throws IOException { FileWriter writer = new FileWriter("test1.txt"); ... writer.write("Zeile 1"); writer.write(System.getProperty("line.separator")); writer.write("Zeile 2"); ... writer.flush(); writer.close(); } }</pre>	
Eingabe (mit den Klassen Scanner und FileReader)	<pre>import java.io.*; import java.util.*; public class Beispiel_09 { public static void main(String[] args) throws IOException { String s; Scanner input = new Scanner(newFileReader("test1.txt")); while(input.hasNext() != false){ s = input.nextLine(); ...; } input.close(); } }</pre>	
5. Verschiedenes		
Formatierte Ausgabe	<pre>System.out.printf(„%20f %20f %20f“, u, v, i); // %f Kommazahlen %d Ganze Zahlen %s Zeichenketten // \n neue Zeile \t Tabulator .x Nachkommastellen</pre>	
Zeichen einlesen	<pre>char op; Scanner input = new Scanner(System.in); op = input.next().charAt(0);</pre>	
IP-Adresse aus String extrahieren	<pre>String s = "192.168.0.1"; StringTokenizer st = new StringTokenizer(s, "."); int v1 = Integer.parseInt(st.nextToken()); int v2 = Integer.parseInt(st.nextToken()); int v3 = Integer.parseInt(st.nextToken()); int v4 = Integer.parseInt(st.nextToken());</pre>	