HOPP ITs

Java

Kartenbeschreibung

KARTENBESCHREIBUNG

OPERATOREN	BEDEUTUNG
zahl = 7;	Weist der Variablen »zahl« den Wert 7 zu.
zahl = zahl + 4;	Erhöht den Wert der Variablen »zahl« um 4.
zahl = zahl – 2;	Verringert den Wert der Variablen »zahl« um 2.
zahl = zahl * 3;	Multipliziert den Wert der Variablen »zahl« mit 3.
zahl = zahl / 4;	Dividiert den Wert der Variablen »zahl« durch 4. Beachte: Falls bei der Integer-Division Nachkommastellen entstehen, werden diese abgeschnitten, z. B. 15 / 4 = 3 (15 / 4 = **3**,75).
zahl = 1 + 3 * zahl;	Es gilt die Punkt-vor-Strich-Rechnung. Multipliziert den Wert der Variablen »zahl« erst mit 3 und erhöht den neuen Wert anschließend um 1.
zahl = zahl % 2;	Weist der Variablen »zahl« den Rest beim Teilen des Werts durch 2 zu. Beispiel: 7 / 2 = 3 Rest 1, also 7 % 2 = 1.
wenn »zahl« negativ:	-8 / 5 = -1 Rest -3, also -8 % 5 = -3.

VERGLEICHS-OPERATOREN UND LOGISCHE OPERATOREN	BEDEUTUNG
(zahl == 5)	Prüft, ob die Variable »zahl« den Wert 5 hat und gibt einen booleschen Wert (wahr/falsch) zurück. Weitere Vergleichsoperatoren: <, <=, >, >=, != (ungleich).
((zahl > 0) && (zahl <= 20)) ((zahl < -30) || (zahl > 30))	Mithilfe der logischen Operatoren UND (&&) und ODER (||) können Vergleiche kombiniert werden. Beim Operator UND müssen beide Vergleiche bzw. Aussagen wahr sein, damit dies auch auf die kombinierte Aussage zutrifft. Beim Operator ODER genügt es, wenn mindestens eine der beiden Vergleiche wahr ist.

KURZSCHREIBWEISEN	BEDEUTUNG
zahl ++;	zahl = zahl + 1;
zahl --;	zahl = zahl – 1;
zahl += 3;	zahl = zahl + 3;
zahl -= 5;	zahl = zahl – 5;
zahl *= 2;	zahl = zahl * 2;

BEDINGTE ANWEISUNG UND VERZWEIGUNG	BEDEUTUNG

if (zahl <= 6) {
Anweisung;
}
	Führt die Anweisung in den geschwungenen Klammern nur dann aus, wenn die Bedingung »zahl <= 6« erfüllt (wahr) ist.

if ((zahl > 0) && (zahl < 20)) {
Anweisung 1;
} else {
Anweisung 2;
}
	Führt die Anweisung 1 nur dann aus, wenn beide Bedingungen (»zahl > 0« **und** »zahl < 20«) erfüllt sind. Ansonsten wird Anweisung 2 ausgeführt.

VERWIRRUNGSKARTEN	BEDEUTUNG
zahl >= 5 ; zahl < 12; zahl == 4;	Der boolesche Wert (s. Vergleichsoperatoren) verändert den Wert der Variablen »zahl« nicht.

int x = 5;
int Zahl = 5;
int zah = 5;
	Eine neue Variable wird deklariert und initialisiert. Sie verändert den Wert der Variablen »zahl« nicht.

if (zahl > 40)
zahl = 5 ;
zahl = 3;
	Ohne geschwungene Klammern umfasst die if-Anweisung nur die erste nachfolgende Zeile. Daher wird in diesem Beispiel nur »zahl = 5« bedingt ausgeführt. »zahl = 3« wird unabhängig davon immer ausgeführt.

SCHLEIFEN	BEDEUTUNG

while (Bedingung) {
Anweisung;
}
	Führt die Anweisung aus, solange die Bedingung erfüllt ist.

for (Initialisierung; Bedingung; Endanweisung) {
Anweisung;
}
 	
Beispiel:
for (int i=0; i<=3; i++) {...}
Anweisung;
}
	Bei der **Initialisierung** wird die Variable i deklariert und mit dem Wert 0 initialisiert. Solange die **Bedingung** erfüllt ist, wird die Anweisung ausgeführt. Ansonsten wird die Schleife beendet. Nach dem Ausführen der Anweisung wird die **Endanweisung** (hier: i++) ausgeführt. Im Beispiel wird die Anweisung genau viermal ausgeführt, nämlich für die Werte i=0, i=1, i=2 und i=3.

for (int i=0; i<3; i++) {
for (int j=0; j<4; j++) {
zahl = zahl + 1;
}
}
	Der Code besteht aus zwei geschachtelten for-Schleifen. Innerhalb jeder Iteration der **äußeren Schleife** gibt es eine **innere Schleife**. In jedem Durchlauf der inneren Schleife wird der Wert der Variablen »zahl« um 1 erhöht. Das bedeutet, dass für jede Iteration der äußeren Schleife die innere Schleife den Wert der Variablen »zahl« viermal erhöht.

ARRAY	BEDEUTUNG
int[] array = {19, -7, 12}; zahl = array[1];	Das Array wird deklariert und mit drei Zahlen initialisiert. Die Zeile »zahl = array[1];« weist der Variablen »zahl« den Wert des **zweiten** Elements des Arrays zu. Position | 1 | **2** | 3 | 4 | 5 | ... Index | 0 | **1** | 2 | 3 | 4 | ...

int[] array = new int[8];
for (int i=0; i<8; i++) {...}
  array[i] = i * i;
}
zahl = array[3];
	Ein Array mit acht Elementen wird deklariert.
	Die Schleife füllt das Array mit den Quadraten der Indizes.
	Abschließend wird der Wert der Variablen »zahl« auf den Wert des **vierten** Elements des Arrays gesetzt.