Vorsichtige Programmierer verwenden Inkrement-Operatoren nicht in komplizierteren Ausdrücken

Aufgabe 1.60

a)

Welchen Wert erhält die boolesche Variable z in folgendem Beispiel?

```
int i = 2, j = 5;
boolean z =  (i <= j - 2) & (2 + ++i == j);
```

b)

Was ändert sich, wenn statt der präfixen (++i) die posfixe Inkrementierung (i++) verwendet wird?

Verzweigungen

Blöcke

Anweisungen werden in Java in Blöcken zusammen-gefasst. Der Anfang und das Ende eines Blocks werden mit geschweiften Klammern gekennzeichnet:

```
{                    // Blockanfang
  Anweisung1;
 Anweisung2;
}                    // Blockende
```

Blöcke können ineinander geschachtelt werden d.h. ein Block kann sich innerhalb eines anderen Blocks befinden.

Ein Block begrenzt gleichzeitig den Gültigkeitsbereich der Variablen, die in diesem Block deklariert werden. Außerhalb dieses Blocks existiert diese Variable nicht und es treten Fehler auf, wenn versucht wird, auf sie zuzugreifen. (lokale Variable)

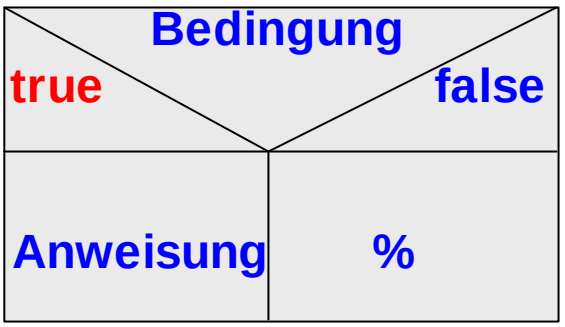
Beispiel:

```
void BlockTest()
{
   int x = 10;        // global für BlockTest
     {                // Beginn des Blocks
     int y = 40;      // lokal für den Anweisungsblock
     y = y + x;
     }                // Ende des Blocks
                      // ab hier kann nur noch auf  x zugegriffen werden
}
```

if-Anweisung

Sie dient dazu, den Ausführungsfluss eines Programms in Abhängigkeit von einer Bedingung zu steuern.

```
if (Bedingung)
  {
   Anweisung1
   Anweisung2
     ......
  }
```



Ergibt die Auswertung der Bedingung *true*, so wird die anschließende Anweisungsfolge ausgeführt.
Besitzt die Bedingung den Wahrheitswert *false*, so wird die Anweisungsfolge nicht ausgeführt, sondern übersprungen.

Die Bedingung *muss* einen Wahrheitswert ergeben. Sie ist in runde Klammern zu setzen.

Beispiel:

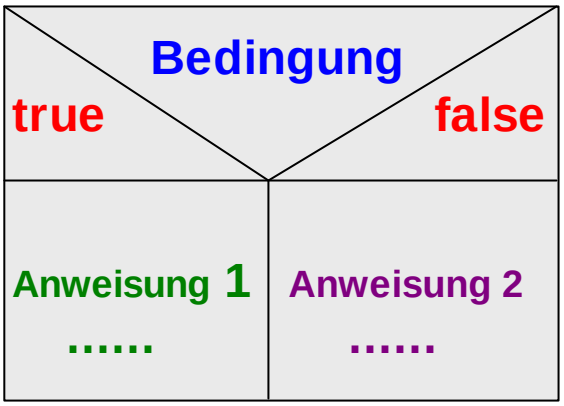
```
import java.util.Random ;

public class wuerfel1{
   public static void main (String args[]){
      Random zzahl=new Random();
      int w=(zzahl.nextInt(6));          // Zufallszahl zwischen -5 und 5
      if (w<0)  w=-w;                    // Zufallszahl zwischen 0 und 5
      w=w+1;                             // Zufallszahl zwischen 1 und 6
      if (w==6)
        {
           System.out.println(" Super, eine 6 gewürfelt");
        }
   }
}
```

if-else-Anweisung

Bei dieser Erweiterung der if-Anweisung wird eine von zwei alternativen Anweisungen ausgeführt.

```
if (Bedingung)
   Anweisungsblock1
else
  Anweisungsblock2
```



Besitzt die Bedingung den Wahrheitswert *true*, wird der *Anweisungsblock1* ausgeführt, andernfalls der *Anweisungsblock2.*

```
import java.util.Random ;

public class wuerfel2{
     public static void main (String args[]) {
       Random zzahl=new Random();
       int w=(zzahl.nextInt(6));
       if (w<0)  w=-w;
       w=w+1;
      if (w==6)
        {  System.out.println(" Super, eine 6 gewürfelt"); }
      else
        {  System.out.println(" Schade, keine 6 gewürfelt"); }
   }
}
```

Hinweis: Da der Anweisungsblock jeweils nur aus einer Anweisung besteht, könnten die Klammern auch entfallen!

Die if-else-Anweisung kann in einfachen Fällen durch den Auswahloperator ? : ersetzt werden:
Bedingung ? Ausdruck1 : Ausdruck2

Beispiel, um das Maximum zweier Zahlen zu ermitteln:

```
int max = (x > y) ? x : y;
```

Falls x > y gilt, wird der Variable max der Wert von x zugewiesen. Andernfalls erhält max den Wert von y.

if-else-Anweisungen können auch ineinander geschachtelt werden.

Beispiel:

```
byte tag;
...
if (tag == 1)
  System.out.println("Mo");
 else if (tag == 2)
      System.out.println("Di");
    else if (tag == 3)
         System.out.println("Mi");
       else if (tag == 4)
            System.out.println("Do");
          else if (tag == 5)
               System.out.println("Fr");
             else if (tag == 6)
                  System.out.println("Sa");
                else if (tag == 0)
                     System.out.println("So");
                   else
                     System.out.println("Kein Tag!");
```

Das Programmstück wird wie folgt ausgewertet: Zunächst wird der Ausdruck tag == 1 ausgewertet. Wenn er den Wahrheitswert true besitzt, so wird "Mo" ausgegeben und die Steuerstruktur verlassen. Ist der Wert false, so wird als nächstes tag == 2 ausgewertet. Abhängig vom Wahrheitswert wird "Di" ausgegeben und "ausgestiegen" oder der nächste Ausdruck ausgewertet. Falls keine der Bedingungen true ist, wird "Kein Tag!" ausgegeben.

Aufgabe 1.80:
Schreibe ein Programm, welches das Minimum zweier übergebener Zahlen ausgibt.

Aufgabe 1.81:
Schreibe ein Programm, das aus der eingegebenen Arbeitszeit den Lohn berechnet. Der Stundenlohn beträgt 14,20€. Arbeitsstunden, die über die ersten 32,5 Stunden hinausgehen, werden mit dem 1,2-fachen Satz entlohnt. Der Stundenlohn ist in einer Konstanten zu speichern.

Aufgabe 1.82:
Schreibe ein Programm zur Lösung liearer Gleichungen (Fallunterscheidungen beachten!)
$mx + b = 0$.

Aufgabe 1.83:
Schreibe ein Programm zur Lösung quadratischer Gleichungen (Fallunterscheidungen beachten!)
$ax^2 + bx + c = 0$.

Aufgabe 1.84:
Schreibe ein Programm Euro_Umrechner.java zur wahlweisen Umrechnung eines DM-Betrags in Euro oder eines Euro-Betrags in DM unter Verwendung einer Konstanten für den Umrechnungsfaktor. Das Programm soll den DM-Betrag bzw. den Euro-Betrag und die Wahl als Parameter übergeben.

Der Aufruf von Euro_Umrechner 1.40 D sollte die Ausgabe 1.40 DM = 0.51 € ergeben.
Der Aufruf von Euro_Umrechner 0.51 E sollte die Ausgabe 0.51 € = 1.40 DM ergeben.

Aufgabe 1.85
Schreiben Sie ein Programm Rechnen2, das mit zwei ganzen Zahlen als Argumenten aufzurufen ist. Das Programm soll die Summe, die Differenz, das Produkt, den Quotienten und den ganzzahligen Rest bei der Division dieser beiden Zahlen ausgeben. Zusätzlich soll überprüft werden, ob der Dividend den Wert 0 hat und entsprechend Reagiert werden.

Der Aufruf Rechnen2 97 0 soll folgende Ausgabe erzeugen:
97+0= 97
97-0 = 97
97*0= 0
97 DIV 0 ist eine nicht definierte Operation
97 MOD 0 ist eine nicht definierte Operation

switch-Anweisung

Die switch-Anweisung erfüllt eine ähnliche Funktion wie die verschachtelte if-else Anweisung, ist aber leichter zu durchschauen:

```
Int i;
switch (i)
 {
        case 1: System.out.println("Mo");
              break;       // Ausstieg aus switch
        case 2: System.out.println("Di");
              break;       // Ausstieg aus switch
        ...
        case 0: System.out.println("So");
              break;       // Ausstieg aus switch
        default: System.out.println("Kein Tag!");
 }
```

Anmerkungen:

- i muss vom Typ char, byte, short oder int sein und in runde Klammern eingeschlossen werden.
- Der Körper von switch besteht aus einem Block.
- Im Innern dieses Blocks stehen eine Reihe von case-Marken und optional eine default-Marke. Diesen Marken folgen Anweisungen ohne Blockklammern.
- Die Werte hinter den case-Marken müssen vom selben Typ wie i sein und müssen alle verschieden sein.
- Die default-Marke darf nur einmal vorkommen.