

1 Standardausgabe auf der Konsole

Mit `System.out.print()` bzw. **`System.out.println()`** lassen sich Daten auf dem Bildschirm ausgeben. Variablen und feste Zeichenketten werden mit „+“ getrennt, was nichts mit der Addition zu tun hat.

5 Anwendungsbeispiele (x sei eine double Variable):

- `System.out.println("Hallo Welt!");`
- `System.out.println("Die Zahl "+x+" ist eine schöne Zahl.");`
- `System.out.println((3*x+5) + " ist das 3fache von "+x+" und noch 3 mehr");`
- `System.out.println("Diese \n Wörter \n stehen \n in \n verschiedenen \n Zeilen");`

10

2 Bedingte Anweisung: if...else

Eine **bedingte Anweisung** ist in der Programmierung eine Anweisung, die nur unter einer bestimmten Bedingung ausgeführt wird. Sie ist einer der wichtigsten Bestandteile der Programmierung, da durch sie ein Programm auf unterschiedliche Zustände und Eingaben reagieren

15 kann.

- wenn (Bedingung) dann Anweisung(en) ansonsten Anweisung(en)

Dies ist folgendermaßen zu lesen: wenn *Bedingung* eintritt, dann wird zu *Anweisung(en)* nur verzweigt.

Nach: http://de.wikipedia.org/wiki/Verzweigung_%28Programmierung%29

20

Syntax allgemein	Beispielmethode
<pre>if (Bedingung) { // Anweisungsblock ... } else { // Anweisungsblock ... }</pre>	<pre>public void prüfeZahl(int pZahl) { if (pZahl>0) { System.out.println(pZahl+" ist positiv."); } else { System.out.println(pZahl+" ist negativ oder 0."); } }</pre>

Syntax allgemein	Beispielmethode
<pre>if (Bedingung) { // Anweisungsblock ... } else { if (Bedingung) { // Anweisungsblock } else { // Anweisungsblock ... } }</pre>	<pre>public void prüfeZahl(int pZahl) { if (pZahl>=10) { System.out.println(pZahl+" ist zwei oder mehrstellig"); } else { if (pZahl<0) { System.out.println(pZahl+"ist negativ."); } else { System.out.println(pZahl+" liegt zwischen 0 und 9.999..."); } } }</pre>

Arbeitshinweise

1. Nennen Sie 3 Zahlen, bei denen jeweils eine andere Zeile die Ausgabe des Programms übernimmt.
Beschreiben Sie, wie die Klammern gesetzt sind, erläutern Sie die Einrückung.
2. Bei dem Beispiel handelt es sich um eine verschachtelte Verzweigung. In einem if...else wird ein weiteres if...else untergebracht.
Legen Sie eine Klasse TestKlasse an. Sie soll keine Eigenschaften haben.
Schreiben Sie eine Methode, die zwei ganze Zahlen als Parameter hat. Die Methode soll die größere auf dem Bildschirm ausgeben. Falls sie gleich groß sind, soll das auch angezeigt werden.

2.1 Bedingte Auszahlung – if...else bei der Online-Bank

Ein konkretes Beispiel: Die Auszahlung erfolgt nur, wenn der es der Kontostand zulässt.

```
25 public void auszahlen(double pBetrag)
   {
     if (kontostand>=pBetrag) {
       kontostand-=pBetrag;
       System.out.println("Auszahlung erfolgt: "+pBetrag);
30     }
     else {
       System.out.println("Auszahlung verweigert!\nEs fehlen"+ (pBetrag-kontostand) );
     }
     System.out.println("Auf Wiedersehen!");
35   }
```

2.2 Eine „is...“-Methode und lokale Variablen

Die folgende Methode prüft, ob der Kontostand positiv ist¹. Typisch ist hierbei, dass Prüfmethode wie diese mit dem Wort „is“, also dem englischen ist, anfangen. Der Rückgabewert der Methode ist vom Typen boolean, d.h. er wird entweder true oder false zurückgegeben.

40 Neu ist die Speicherung von Daten in einer „lokalen Variable“ ergebnis. Diese ist eine lokale Variable, da sie nur innerhalb der Methode isPositiv() gültig ist.

Lokale Variable: Eine Variable, die innerhalb einer Methode angelegt wird, wird lokale Variable genannt. Sie wird mit verlassen der Methode wieder zerstört. Sie kann nur innerhalb der Methode abgefragt und verändert werden.

Nennt man eine lokale Variable genauso wie eine Eigenschaft der Klasse, so muss man zur Unterscheidung das Wort this verwenden, um die Eigenschaft zu erreichen. Bei Namensgleichheit hat das Abfragen der lokalen Variable Priorität.

Lokale Variablen lassen sich gut als kurzfristigen **Zwischenspeicher** einsetzen.

Konvention: Die Deklaration der lokalen Variable sollte direkt unter dem Kopf der Methode – also direkt nach der ersten geschweiften Klammer - passieren.

```
45 public boolean isPositiv()
   {
     boolean ergebnis;
     if (kontostand>0) {
       ergebnis=true;
     }
     else {
50       ergebnis=false;
     }
   }
```

1 Eigentlich ist 0 sowohl positiv als auch negativ. Das berücksichtigt die Methode aber nicht. Ggf. müsste man die Methode isGroesserNull() nennen.

```
return ergebnis;  
}
```

Arbeitshinweise

1. Fügen Sie eine Eigenschaft Dispo-Kredit mit einer get- und set-Methode ein. Ändern Sie anschließend die Methoden abheben() so, dass der Dispo-Kredit nicht überschritten werden kann.
2. Durch Einzahlen eines negativen Betrages kann man den Bankautomaten austricksen. Ändern Sie diese Sicherheitslücke (umgehend ;-)).
3. Erweitern Sie das Konto um eine Eigenschaft „gesperrt“. Implementieren Sie die zugehörige Methode isGesperrt(), die true zurückgibt, wenn das Konto gesperrt ist.

2.3 Links

- http://de.wikibooks.org/wiki/Java_Standard:_Kontrollstrukturen

2.4 BMI-Rechner

„Der **Körpermasseindex (KMI)** - oft auch: *Body-Mass-Index (BMI)*, *Kaup-Index* oder *Körpermassenzahl (KMZ)* - ist eine Maßzahl für die Bewertung des Körpergewichts eines Menschen im Verhältnis zum Quadrat seiner Größe. Sie wurde von [Adolphe Quetelet](#) entwickelt. Da [Übergewicht](#) ein weltweit zunehmendes Problem darstellt, wird die Körpermassenzahl vor allem dazu verwendet, auf eine diesbezügliche Gefährdung hinzuweisen.

Der BMI gibt lediglich einen groben Richtwert an und ist nicht unumstritten, da er die [Statur](#) eines Menschen und die individuell verschiedene Zusammensetzung des Körpergewichts aus [Fett-](#) und [Muskelgewebe](#) naturgemäß nicht berücksichtigt. Ebenso ist er für besonders große und besonders kleine Menschen nur eingeschränkt aussagefähig.“

Arbeitshinweise

1. Entwerfen Sie die Klasse Person, die das Alter, das Gewicht und die Größe einer Person erfasst. Entwickeln Sie die Methode getBMI(), die den BMI berechnet und zurückgibt.
$$\text{Körpermassenzahl} = \frac{\text{Masse}}{\text{Größe}^2}$$
2. Lassen Sie prüfen, ob die Person Unter-, Normal- oder Übergewicht hat. Untergewicht soll ein Wert unter 20, Übergewicht ein Wert über 25 sein.
Fortgeschrittene: Realisieren Sie die gesamte Tabelle (kritisches Untergewicht und „Adipositas“, d.h. Starkes Übergewicht über BMI 30)
3. Lassen Sie prüfen, ob die Person ihr Idealgewicht hat. Es soll true zurückgegeben werden, wenn das der Fall ist.
(Hinweis: Es ist einfacher mit && zu arbeiten als mit if...else. Fragen Sie Ihren Lehrer!)
4. Realisieren Sie eine Klasse Zahlenratespiel. Eine Zahl zuErratendeZahl wird als Eigenschaft vom Konstruktor gesetzt (z.B. auf 7).
a) Entwickeln Sie eine Methode, rateZahl(int zahl), die zahl einliest und mit zuErratendeZahl zahl vergleicht. Lassen Sie ausgeben „zu klein“, zu groß“ oder „Gewonnen!“.
b) Erweitern Sie die Methode um eine Abfrage: der Ratende soll nur Zahlen zwischen 1 und 10 eingeben können.

<http://de.wikipedia.org/wiki/Body-Mass-Index>

Bei der Interpretation des BMI ist das **Alter** einer Person zu berücksichtigen.

Altersabhängiges **Normalgewicht**:

Kategorie	BMI (kg/m ²)
kritisches Untergewicht	< 17
Untergewicht	17–20
Normalgewicht	20–25
Übergewicht	25–30

Alter (Jahre)	BMI-Idealwert (kg/m ²)
19–24	19–24
25–34	20–25
35–44	21–26
45–54	22–27
55–64	23–28
> 64	24–29

2.5 Logische Operatoren NICHT, UND und ODER

Wichtige logische Operatoren sind NICHT, UND und ODER. Wir verwenden sie alltäglich, wenn wir Aussagen formulieren. Allerdings sind wir im Alltag häufig unpräzise. Im Alltag verwenden wir Logische Operatoren ohne es zu bemerken häufig bei Sätzen mit „Wenn...“:

- Wenn ich eine gute Note in der Klausur habe UND ich im Unterricht mitmache, dann bekomme ich eine gute Note (d.h. nur wenn beides der Fall ist).

In Java: `if ((klausurNote<=2) && (soMiNote <=2)) ...`

- Wenn es regnet ODER die Sonne stark scheint, sitze ich unter einem Schirm (d.h. wenn **zumindest eines erfüllt** ist. Wenn es regnet und zugleich die Sonne stark scheint, sitze ich natürlich erst recht unter einem Schirm.)

In Java: `if ((sonneScheint==true) || (esRegnet==true)) ...`

- Wenn es NICHT hell ist, dann geht die Straßenlaterne an.

In Java: `if (!(helligkeitsSensor>30)) ... ist das gleiche wie: if (helligkeitsSensor<=30) ...`

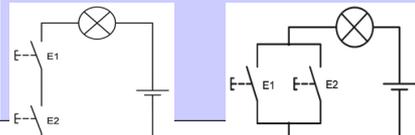
Zusammenfassung: Für zwei boolesche Werte (also bei Java Typ boolean) a und b gilt:

- **!a** (bedeutet NICHT a) erzeugt das Gegenteil von a. Wenn a true ist, ist !a false und andersherum.
- Die Aussage **a && b** (bedeutet a UND b) ist true, wenn **sowohl a als auch b** true sind. Sonst nicht.
- Die Aussage **a || b** ist true, wenn **entweder a oder b oder beide** true sind. D.h. nur wenn sowohl a als auch b false sind, ist die Aussage a || b auch false.

Übung:

1) Nennen Sie weitere Beispiele, wo logische Aussagen verwendet werden und prüfen Sie, ob es mit der Logik übereinstimmt.

2) Überprüfen Sie die folgenden Aussagen:
 Wenn (a || b) true ist, weiß ich, dass mindestens eine der Variablen true ist. (wahr/falsch)
 Wenn (a && b) true ist, weiß ich, dass jede der Variablen true ist. (wahr/falsch)
 Wenn (a && b) false ist, weiß ich, dass jede der Variablen false ist. (wahr/falsch)
 Wenn (a || b) false ist, weiß ich, dass jede der Variablen false ist. (wahr/falsch)
 Wenn !(a && b) false ist, weiß ich, dass keine der Variablen true ist. (wahr/falsch)
 Wenn !(a || b) false ist, weiß ich, dass beide Variablen true sind. (wahr/falsch)
 (a && b) || !(a && b) ist immer true. (wahr/falsch)

3)  Die Schaltungsskizzen links charakterisieren logische Operatoren. Erläutern Sie den Zusammenhang.

Übung:	
aus Wikipedia (Die genaue Quellenangabe resultiert aus Lösung der Aufgabe und wird deshalb nicht genannt)	Vertiefung: Schlagen Sie in Wikipedia nach: <ul style="list-style-type: none">● Aussagenlogik● Negation (Komplement-Gatter)● Konjunktion (Und-Gatter)● Disjunktion (Oder-Gatter)

2.6 Anwendung: Die kleinste aus 3 Zahlen.

Drei Zahlen werden als Parameter übergeben, die kleinste soll zurückgegeben werden. Die folgende Methode liefert einen Lösungsansatz, der noch nicht vollkommen optimal ist (vgl. Aufgabe 2).

35

40

```
public int kleinsteZahl(int x, int y, int z)
{
    int zwischenspeicher=x;
    if ((x<y) && (x<z)) zwischenspeicher = x;
    if ((y<x) && (y<z)) zwischenspeicher = y;
    if ((z<x) && (z<y)) zwischenspeicher = z;
    return zwischenspeicher;
}
```

Übung:	
1)	Analysieren Sie die Methode und erläutern Sie dabei die Anwendung des logischen Unds.
2)	Wie reagiert die Methode auf die Eingabe von zwei gleich großen Zahlen y und z und einem größeren x. Prüfen Sie dies mit Hilfe eines Schreibtischtestes. Schreiben Sie die Methode so um, dass auch bei gleich großen Zahlen zumindest eine der kleinsten Zahlen ausgegeben wird.
3)	Realisieren Sie eine Variante für 4 Zahlen. Schätzen Sie ab, wie aufwändig eine Variante mit 5, 6, 7... Zahlen ist.
4)	Ingenieur Meyer hat einen Bankautomaten konzipiert, bei dem er sich ein Hintertürchen offen gelassen hat: Variante a) Eine Auszahlung ist möglich, wenn der Kontobesitzer Meyer heißt, auch wenn der Betrag nicht mehr verfügbar ist. Aus Angst, aufzufliegen, da es zu viele Meyers gibt, hat er folgende Variante realisiert: Variante b) Eine Auszahlung ist möglich, wenn der Kontobesitzer Meyer aber auch nur genau dann, wenn der Abhebebetrag 650 Euro ist, egal ob der Betrag verfügbar ist. Ergänzen Sie beide Varianten in Ihrer Onlinebank und diskutieren Sie über den Realismus dieser Aufgabe. Hinweis: Statt == zu verwenden prüft man Strings mit <code>if (name.equals("Mayer")) ...</code>