

Geradeausprogramme (Sequenz)

Geradeausprogramme entstehen durch Aneinanderreihung von Anweisungen (Wertzuweisung, Eingabe- und Ausgabeanweisung usw.) zu einer Sequenz (Folge von Anweisungen).

Aufgabe 1:

Erstellen Sie ein Programm, das nach Eingabe zweier beliebiger ganzer Zahlen x und y vom Typ `int` die Ergebnisse der Operationen $+$, $-$, $*$, $/$, $\%$, $++$, $--$ wie folgt auf dem Bildschirm ausgibt. Hinweis: Speichern Sie den zu Beginn eingegebenen Wert der Variablen x in einer Hilfsvariablen, damit sich die Werte von $++x$, $--x$, $x++$ und $x--$ immer auf diesen beziehen.

OPERATIONEN MIT GANZEN ZAHLEN

```
Bitte zwei ganze Zahlen eingeben:  x =
                                     y =
x + y =
x - y =
x * y =
x / y =
x % y =
++x  =
--x  =
x++  =
x--  =
```

Formatierte Ausgabe

Escape-Sequenzen

Unter Escape-Sequenzen versteht man die Kodierungen von unerlaubten Zeichen oder Formatierungen mittels eines reservierten Steuerzeichens. Das ist bei C++ der Backslash (`\`).

Escape-Sequenz	Zeichen / Formatierung
<code>\b</code>	Rückschritt (backspace)
<code>\n</code>	Zeilenvorschub (newline)
<code>\r</code>	Zeilenrücklauf (carriage return)
<code>\t</code>	Horizontaler Tabulator (tab)
<code>\a</code>	Gong-Zeichen
<code>\'</code>	Einfaches Anführungszeichen (apostroph)
<code>\"</code>	Doppeltes Anführungszeichen
<code>\\</code>	Schrägstrich (backslash)

Aufgabe 2:

Erstellen Sie vom Programm aus Aufgabe 1 eine Kopie. Ändern Sie die Kopie so ab, dass die Ausgabe auf dem Bildschirm jetzt tabellarisch erfolgt und in der Überschrift Anführungszeichen vorkommen:

OPERATIONEN MIT "GANZEN ZAHLEN"

```
Bitte zwei ganze Zahlen eingeben:  x =
                                     y =
x + y =           x - y =           x * y =
x / y =           x % y =           ++x  =
--x  =           x++  =           x--  =
```

Manipulatoren

Ohne zusätzliche Angaben werden durch *cout* (Objekt der Klasse *ostream*) Daten im Standardformat des jeweiligen Datentyps auf dem Bildschirm ausgegeben. Zahlen werden dezimal ausgegeben, Gleitpunktzahlen im Format 123.456 oder 1.234560e+002 je nachdem was kürzer ist. Zur Formatierung der Ausgabe stehen in der Bibliothek *iomanip.h* so genannte *Manipulatoren* zur Verfügung:

Manipulator	Bedeutung	beeinflusst
<i>dec</i>	dezimale Ausgabe (Voreinstellung)	ganzzahlige Werte
<i>hex</i>	hexadezimale Ausgabe	ganzzahlige Werte
<i>oct</i>	oktale Ausgabe	ganzzahlige Werte
<i>setw(n)</i>	<i>n</i> (Typ <i>int</i>) ist die minimale Ausgabenbreite	alle außer <i>char</i>
<i>setprecision(n)</i>	<i>n</i> (Typ <i>int</i>) ist die Ausgabegenauigkeit	Gleitkommawerte

Aufgabe 3:

a) Testen Sie folgendes Programm. Wie werden die Daten ausgegeben?

```
//format.cpp
#include <iostream.h>
#include <iomanip.h>
int main()
{
    int x = 1234;
    cout << setw(15) << "dezimal:" << setw(10) << dec << x << endl
         << setw(15) << "hexadezimal:" << setw(10) << hex << x << endl << endl;
    return 0;
}
```

b) Ergänzen Sie das Programm durch die oktale Ausgabe von *x*.

Aufgabe 4:

Schreiben Sie ein Programm, das nach Eingabe einer beliebigen Zahl (12.345 ist nur ein Beispiel) vom Typ *double* folgendes (formatiert!) ausgibt (Hinweis: in der Bibliothek *math.h* gibt es die Funktionen *sqrt(x)* und *pow(x, y)* zur Berechnung von Quadratwurzeln und Potenzen).

Bitte beliebige Gleitpunktzahl eingeben: 12.345

```

    Zahl                12.345
    Quadratzahl        152.399
    Kubikzahl          1881.37
    Quadratwurzel      3.51355
```

Um eine Gleitpunktzahl *x* mit einer festen Anzahl *n* von Nachkommastellen auszugeben, müssen folgende Anweisungen ausgeführt werden:

```
cout.setf(ios::fixed);
/*setf ist eine Memberfunktion der Klasse ostream; ios ist Basisklasse von ostream und
istream; fixed ist Memberfunktion von ios*/
cout << setprecision(n);
cout << x;
```

Aufgabe 5:

Ergänzen Sie das Programm aus Aufgabe 4 so, dass alle Zahlen mit genau drei Stellen nach dem Komma ausgegeben werden.

Ausführung von DOS-Befehlen

Die Funktion *system* aus *<stdlib.h>* übergibt eine Zeichenkette an den Kommandointerpreter von DOS zur Ausführung des entsprechenden Befehls.

system("cls"); bewirkt z. B., dass der DOS-Befehl *cls* (clear screen) ausgeführt wird.