

# Einführung in die Programmierung

## Lernzielkontrolle:

In dieser Lernzielkontrolle wird folgendes überprüft:

- Die Verwendung einfacher Datentypen (int, double, long...).
- Die Anwendung der einfachen Verzweigung (if-Anweisung).
- Die Anwendung der mehrfachen Verzweigung (switch-Anweisung).
- Die Erstellung und Darstellung einfacher Algorithmen (Struktogramm/Flussdiagramm).



### Aufgabe 1/15 ✓

**Verzweigung** – Schreiben Sie ein Programm, das drei Werte x, y, und z entgegennimmt und prüft, ob genau 2 Werte gleich sind. Kommentieren Sie das gesamte Programm.



### Aufgabe 2/15 ✓

*Ausgabe ist falsch*

**Verzweigung** – Schreiben Sie ein Programm, das drei Zahlen a, b und c entgegennimmt und sie in absteigender sortierter Reihenfolge wieder ausgibt. Kommentieren Sie dabei alle *then* und *else* Zweige.



### Aufgabe 3/15 ✓

**Mehrfach Verzweigung** – Schreiben Sie ein Programm, das die Monatstage in Abhängigkeit vom Monat ermittelt (ohne Schaltjahr). Kommentieren Sie dabei alle *case* Pfade.

31: 1  
30: 4  
2  
5  
8  
10  
12  
28: 2



### Aufgabe 4/20

**Codevereinfachung** – Vereinfache folgendes Codefragment:

```
if (x < 0 && y < 0)
    a = x * y;
else if (x < 0)
    a = x * (-y);
else if (y > 0)
    a = (-x) * (-y);
else
    a = x * (-y);
```

wenn x & y negativ

wenn nur x negativ

wenn y positiv

*falls x < 0*

*a = x \* (-y)*

### Aufgabe 5/15 ✓

**Schleifen** – Welche der folgenden Schleifen terminiert (Begründung)?

#### Schleife A

```
int y = 1, j = 1;
do {
    y = y + j;
    j--;
} while(y < 200);
```

#### Schleife B

```
int x = 1, j = 20;
while (x + j < 100) {
    x = x + 3;
    j--; j--;
```



### Aufgabe 6/20 ✓

**Schleifen** – Schreiben Sie ein Programm, das die Ziffernsumme einer positiven ganzen Zahl n berechnet. (Struktogramm/Flussdiagramm). Für die Zahl 4711 beträgt die Ziffernsumme 13.

1)

```

double x,y,z; // Variablen deklarieren
if(x==y){ // x & y auf Gleichheit prüfen genau
    Console.WriteLine($"{{x}} & {{y}} sind gleich"); // Ausgabe 2 Werte
}
if(y==z){ // y & z prüfen
    Console.WriteLine($"{{y}} & {{z}} sind gleich"); // Ausgabe
}
if(z==x){ // z & x prüfen
    Console.WriteLine($"{{z}} & {{x}} sind gleich"); // Ausgabe
}
else{ // Wenn nichts gleich ist:
    Console.WriteLine("Keine der Zahlen sind gleich!"); // Ausgabe
}

```

2)

```

double a,b,c,swap; // Absteigend
if(a < b){ // Wenn a < b → tausche
    swap = a; b = a; a = swap;
}
if(a < c){ // Wenn a < c → tausche
    swap = a; c = a; a = swap;
} // a nun garantiert höchster Wert
if(b < c){ // 
    swap = b; c = b; b = swap;
}
Console.WriteLine($"größter Wert: {{a}} in Mitte: {{b}} In kleinster: {{c}}")

```

3)

```
int month; days; ✓  
switch(month) {
```

```
case 1: // Januar  
case 3: // März  
case 5: // Mai  
case 7: // Juli  
case 8: // August  
case 10: // Oktober  
case 12: // Dezember = 31 Tage  
days = 31;  
break;
```

```
case 4: // April  
case 6: // Juni  
case 9: // September  
case 11: // November = 30 Tage  
days = 30;  
break;
```

```
case 2: // Februar = 28 Tage  
days = 28;  
break;
```

```
default: // Sollte nicht erreicht werden,  
break; (wenn ungültiger Monat eingegeben wurde)
```

```
Console.WriteLine($"Der Monat {month} hat {days} Tage");
```

5) loop A:

nach 1. 2. 3. Durchlauf: 4. Durchlauf:

$$\begin{array}{llll} y = 2 & y = 2 & y = 1 & y = -1 \\ j = 0 & j = -1 & j = -2 & j = -3 \\ & & & \dots \end{array}$$

A: A terminiert NICHT! (die Bedingung  $y < 200$  ist  
A ist eine Endlosschleife. immer erfüllt)

loop B:

nach 1. 2. 3. 4. Durchlauf:

$$21 < 100 \quad 22 < 100 \quad 23 < 100 \quad 24 < 100$$

$$\begin{array}{llll} x = 4 & x = 7 & x = 10 & x = 13 \\ j = 18 & j = 16 & j = 14 & j = 12 \\ & & & \dots \end{array}$$

A: B terminiert!

(nach  $n$  Durchläufen ist die Bedingung  $(x+j < 100)$   
nicht weiter erfüllt.., da  $x$  um 3 wächst, während  
 $j$  jeweils um 2 schrumpft..)

6) ✓ int n, sum; //zB: 123

sum = 0;

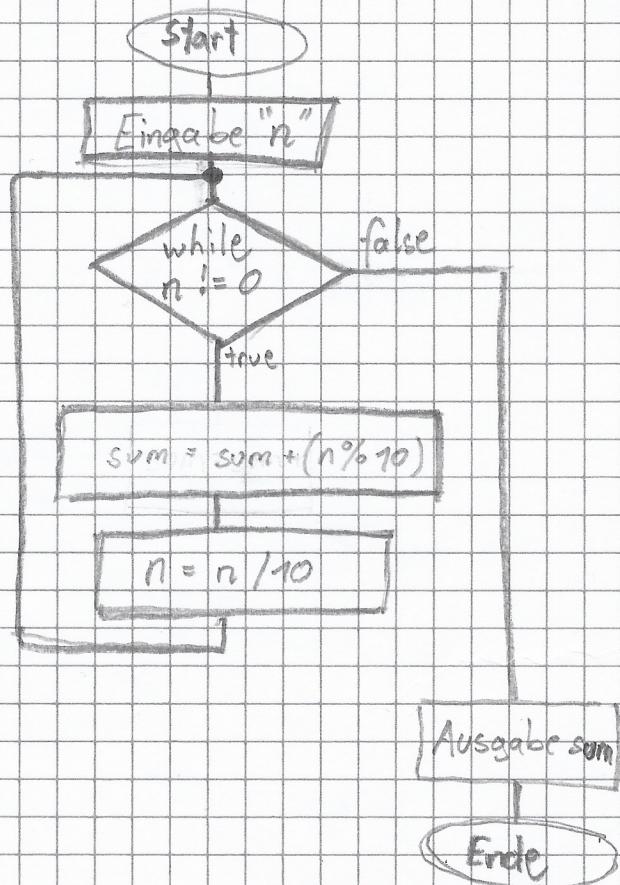
while (n > != 0) {

    sum = sum + (n % 10);      sum = 3      sum = 5      sum = 6

    n = n / 10;      n = 12      n = 1      n = 0

}

Console.WriteLine("die Ziffernsumme der Zahl ist Esom");



$$4) \quad x = -1 \quad y = -1 \quad \rightarrow \quad a = 1$$

$$x = -1 \quad y = 1 \quad \rightarrow \quad a = -1$$

$$x = 1 \quad y = -1 \quad \rightarrow \quad a = 1$$

$$x = 1 \quad y = 1 \quad \rightarrow \quad a = 1$$

if ( $y > 0$ )

↓  
≠

$a = x * (-y);$

else

$a = x * y;$

~~$a = x * (y > 0) ? (x * (-y)); (x * y);$~~