

### Arbeitsblatt 3. Verzweigungen, Schleifen

Die Syntax zu Verzweigungen und Schleifen finden Sie in den entsprechenden Handouts im Git: <https://git.informatik.fh-nuernberg.de/albrecht-prog1/ws2024/Handouts>

#### Aufgabe 3.1. Beitragsrechnung für einen Verein

Erstellen Sie ein Programm, das den Vereinsbeitrag für eine Person berechnet, abhängig vom Alter und dem sozialen Status. Es gelten folgende Beitragssätze:

0 bis 6 Jahre: frei

7-17 Jahre: 30 €

18-65 Jahre: Wenn erwerbslos 40€, ansonsten 80€

ab 66 Jahre: 50€

Mögliche Abläufe dieses Programmes:

```
Beitragsrechnung für einen Verein
=====
Geben Sie Ihr Alter ein: 14
Ihr Beitrag ist 30 Euro.
```

```
Beitragsrechnung für einen Verein
=====
Geben Sie Ihr Alter ein: 38
Sind Sie erwerbslos (j/n): j
Ihr Beitrag ist 40 Euro.
```

```
Beitragsrechnung für einen Verein
=====
Geben Sie Ihr Alter ein: 72
Ihr Beitrag ist 50 Euro.
```

#### Aufgabe 3.2. For-Schleifen

- a) Geben Sie die int-Zahlen von 3..77 aus.
- b) Geben Sie alle Vielfachen von 5 zwischen 0..1000 aus.
- c) Geben Sie in Dreierschritten von 100 abwärts jede positive Zahl aus.

#### Aufgabe 3.3. Quadratzahlen

Schreiben Sie ein Programm, das eine Zahl einliest und die Folge der Quadratzahlen ausgibt, solange die Quadratzahl kleiner oder gleich diese Zahl ist.

Welche Form von Schleife ist dafür am besten geeignet?

Überlegen Sie sich vorab die Schleifen-Logik (vielleicht mit einem PAP) in dieser Reihenfolge:

1. Was für Variablen benötigen Sie? Wie müssen sie initialisiert werden?
2. Was muss in der Schleife passieren?
3. Wie ist ihre Abbruch-Bedingung?

Beispielausgabe:

```
Zahl eingeben: 120
Quadratzahlen bis max. 120: 1, 4, 9, 16, 25, 36, 49, 64, 81, 100
```

## Prozedurale Programmierung

Optional: Versuchen Sie, die Ausgabe genauso aussehen zu lassen, wie in diesem Beispiel, d.h. finden Sie eine Lösung für das letzte Komma.

### Aufgabe 3.4. Rotation

Schreiben Sie ein C#-Programm, das für drei Variable a, b, c Werte von der Konsole einliest, die Werte dieser Variablen dann um eine Stelle rotiert und wieder ausgibt. Nach der Rotation sollen der ursprüngliche Wert

Achtung: Es sollen die neuen Werte wirklich in den Variablen gespeichert sein und nicht nur in der Ausgabe vertauscht! Brauchen Sie dafür eine Schleife oder eine Verzweigung oder vielleicht gar nichts davon?

Beispiel für einen möglichen Programmablauf:

Bitte geben Sie drei Zahlen ein:

5  
6  
7

Vor Rotation:

a=5, b=6, c=7

Nach Rotation:

a=6, b=7, c=5

Kopieren Sie sich diesen Code und passen, ergänzen Sie das Nötige bei "Hier Rotation durchführen", aber lassen Sie den Rest so, wie er ist.

```
Console.WriteLine("Bitte geben Sie drei Zahlen ein:");

int a = Convert.ToInt32(Console.ReadLine());
int b = Convert.ToInt32(Console.ReadLine());
int c = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Vor Rotation:");
Console.WriteLine($"a = {a}, b = {b}, c = {c}");

// Hier die Rotation durchführen

Console.WriteLine("Nach Rotation:");
Console.WriteLine($"a = {a}, b = {b}, c = {c}");
```

### Aufgabe 3.5. Temperatur-Umrechner

Es gilt die Formel: Grad Fahrenheit =  $(9/5 * \text{Grad Celsius}) + 32$

Schreiben Sie ein Programm, das eine Umrechnungstabelle für die Temperaturen zwischen 0 und 100 °C in Vierer-Schritten ausgibt. Die Formatierung ist nicht wichtig, aber wenn Sie Zeit haben, lassen Sie die Ausgabe wie in dem Beispiel aussehen.

°C	°F
0,0	32,0
4,0	39,2
8,0	46,4
...	...
92,0	197,6
96,0	204,8
100,0	212,0