

# Programmieren Tutorium

## Arbeitsblatt 6

Datum: 5. Dezember 2024

### Aufgabe 1: Klausuraufgaben aus Programmieren I WS 2018

Übersetzen Sie die folgenden kursiv gedruckten Sätze in C#-Anweisungen.

- a) Die Variablen `i1` und `i2` enthalten ganze Zahlen. *Wenn die Differenz aus `i1` und `i2` kleiner oder gleich 2 ist, dann wird die Hälfte von `i1` ausgegeben. Andernfalls wird `i2` verdoppelt.*
- b) Die Variable `d1` ist eine Gleitpunktzahl. *Solange `d1` größer als 5 ist, wird das gemacht: Der Wert von `d1` wird um 3 vermindert. Wenn dann das Quadrat von `d1` kleiner als 10 ist, wird das Dreifache von `d1` ausgegeben.*
- c) Die Variable `t1` ist vom Typ `string`, anz von Typ `int`. Übersetzen Sie: *Für jedes Element des Strings wird das gemacht: Wenn der ASCII-Wert des Elements gerade ist, wird der Wert von `anz` um 3 erhöht.*
- d) Die Variable `iifeld` ist vom Typ `List<int>`. Übersetzen Sie: *Für alle Elemente von `iifeld` vom ersten bis zum letzten wird das gemacht: Wenn das aktuelle Element eine Zahl zwischen 65 und 91 (beides einschließlich) ist, wird der Buchstabe ausgegeben, dessen ASCII-Wert dem Zahlenwert entspricht.*

**Hinweis:** Mit `(char) <INTEGER>` bekommst du den ASCII Buchstaben von einer Zahl. z.B `(char) 65 => A`.

- e) Die Variable `bfield` ist vom Typ `bool[]`. *Für jedes fünfte Element von `bfield` wird das gemacht: Wenn der Index durch 2 teilbar ist, wird das Element auf `true` gesetzt, sonst auf `false`.*

### Aufgabe 2: structs/Klassen definieren

**Hinweis:** In C# wird ein `struct` wie folgt definiert:

```
public struct NameDesStructs {  
    public TYP AttributName1;  
    public TYP AttributName2;  
}
```

**Hinweis:** In C# wird eine Instanz/Objekt einer Klasse/struct wie folgt erstellt:

```
var variablenName = new NameDesStructs();
```

Oder so:

```
NameDesStructs variablenName = new NameDesStructs();
```

a) Definiere eine Struktur/struct/Klasse `Person`. Erstelle dann zwei Personen-Objekte/Instanzen.

b) Füge ein Feld `Name` zur Struktur/struct/Klasse `Person` hinzu. Erstelle dann zwei Personen Objekte und weise ihnen verschiedene Namen zu. Gib die Namen der beiden Personen auf der Konsole aus.

c) Struct `Person` mit `Name` und `Alter`:

Erweitere die Struktur `Person` um ein weiteres Feld `Alter`. Erstelle ein `Person`-Objekt mit dem Namen `Hänsünd` einem Alter von 20. Simuliere einen Geburtstag, indem du das Alter um 1 erhöhst, und gib das Alter vor und nach dem Geburtstag auf der Konsole aus.

d) Vergleich von zwei Personen nach Alter.

Verwende die Struktur `Person` mit den Feldern `Name` und `Alter`. Frage den Namen und das Alter von zwei Personen ab. Vergleiche das Alter der beiden Personen und gib aus: Person X ist älter als Person Y oder Person X und Y sind gleich alt. (Entweder direkt in der Main oder schreibe eine einfache statische Methode

e) Liste von Personen (anspruchsvoll)

**Hinweis:** In C# wird eine Liste von Personen wie folgt erstellt

```
var personenListe = new List<Person>();
```

Erstelle eine leere Liste von Personen. Füge anschließend zwei Personen zur Liste hinzu: Martin mit dem Alter 22 und Julian mit dem Alter 40. Gib das Alter der Person an Position 0 in der Liste, auf der Konsole aus.

f) Suche in der Liste (schwer)

Du hast eine Liste von Personen. Schreibe eine statische Funktion, die überprüft, ob eine Person mit einem bestimmten Alter X in der Liste existiert.

g) Verbessertes Personen erstellen (schwer):

Schreibe eine Konstruktor für `Person`. An diesen soll der Name und das Alter übergeben werden.

**Hinweis:** In C# definiert man einen Konstruktor folgendermaßen:

```
public struct Person{
    // Attribute
    public Person(TYP parameterName1, TYP parameterName2){
        // Initialisieren des Objekts, Erstellen des Objekts
    }
}
```

h) Personen ausgeben (schwer):

Implementiere eine ToString Methode in der Klasse Person. Diese soll ausgeben: Hans ist 18 Jahre alt.

**Hinweis:** Um die Ausgabe eines Objekts zu verändern, definiert man die ToString Methode

```
public struct Person{
    public override string ToString(){
        // Wichtig du musst einen String returnen, keine
        // Console.WriteLine Statements
        // Meistens gibt man die Attribute an! Z.B
        return $"Person: Vorname {this.FirstName}"
    }
}

class Program{
    public static void Main(){
        var person = new Person("Hans", 20);
        Console.WriteLine(person);
    }
}
```

## Aufgabe 3: structs und Funktionen

a) Struct Handy mit Ladestand Erstelle eine Struktur Handy mit einem Feld Akkustand, der den aktuellen Ladestand in Prozent von 0 bis 100 speichert. Erstelle ein Handy-Objekt und setze den Akkustand auf 100.

b) Methode zum Aufladen des Akkus

Schreibe eine Methode AkkuAufladen, die den Akkustand des Handys um 1% erhöht.

c) Verhindere, dass der Akkustand über 100% steigt, indem die Funktion AkkuAufladen aus b) bei 100% keine weitere Erhöhung vornimmt.

d) Liste von Handys Erstelle eine Liste von Handys und füge 3 Handy-Objekte mit verschiedenen Akkuständen hinzu. Lade das Handy an Index 2 der Liste auf. Gebe den Akku Stand des Handys vor und nach dem Laden aus.

**Hinweis:** Erstellen einer Handy Liste:

```
var handys = new List<Handy>();
var handy1 = new Handy();
handy1.Akku = 20;
```

e) Erstellen von 100 Handys mit unterschiedlichen Akkuständen (anspruchsvoll)

Erstelle eine Liste mit 100 Handy-Objekten. Das erste Handy soll einen Akkustand von 0 haben, das zweite 1, das dritte 2 usw., bis zum letzten Handy mit einem Akkustand von 99.

f) Schreibe einen Konstruktor und ToString (schwer)

Überlege dir einen sinnvollen Konstruktor und eine ToString MEthode für das Handy.

d) Größter Akkustand in der Liste (schwer):

Finde den größten Akkustand in der Liste von Handys und gib den Akkustand des Handys mit der höchsten Ladung aus.

## Aufgabe 4: Erstelle eine Anwesenheitsliste

Erstelle ein Anwesenheitsliste struct. Das struct soll eine Liste an Personen als Attribut haben.

- a) Schreibe einen Konstruktor, welcher keine Argumente/Parameter übergeben bekommt
- b) Schreibe eine personHinzufügen(Person person) Methode/Funktion
- c) Schreibe eine istPersonAnwesend(Person person) Methode/Funktion
- d) Schreibe eine berechneAnzahlAnwesendePersonen() Methode/Funktion