

Programmieren Tutorium

Arbeitsblatt 7

Datum: 5. Dezember 2024

Aufgabe 1: Wie Klausuraufgabe 1

Übersetzen Sie die folgenden Sätze in C#-Anweisungen.

- a) Die Variablen z1 und z2 enthalten ganze Zahlen. Übersetzen Sie: Wenn z2 kleiner oder gleich z1 und außerdem z2 größer als 4 ist, wird z1 der Kehrwert von z2 zugewiesen (also 1 geteilt durch z2). Dabei werden die Nachkommastellen abgeschnitten.
- b) Die Variable st ist ein String. Übersetzen Sie: So lange die Länge von st nicht durch 7 teilbar ist, wird ein Fragezeichen an den String angehängt
- c) Die Variable df ist ein dynamisches Feld von Gleitpunktzahlen. Übersetzen Sie: Für alle Zahlen des Feldes wird das Vierfache des Feldelementes ausgegeben.
- d) Die Variable sfeld ist ein Stringfeld vom Typ List<string>. Übersetzen Sie: Für jedes Element des Feldes wird das gemacht: Wenn die Länge des Elements größer als 5 ist wird an das Element ein Punkt angehängt.

Aufgabe 2: User - Konstruktor - To String - Properties

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 User. Erstelle dann zwei User-Objekte/Instanzen.
- b) Füge ein Feld Vorname zum struct User hinzu. Erstelle dann zwei User Objekte und weise ihnen verschiedene Vornamen zu. Gib die Vornamen der beiden Personen auf der Konsole aus.
- c) Füge ein Feld Nachname zum struct User hinzu.
- d) Verbessertes User erstellen:
Schreibe eine Konstruktor für User. An diesen soll der Vor und Nachname übergeben werden.

Hinweis: In C# definiert man einen Konstruktor folgendermaßen:

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

e) Personen ausgeben:

Implementiere eine ToString Methode in der Klasse Person. Diese soll ausgeben: User: Udo Müller.

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);
    }
}
```

f) GanzerName Property:

Implementiere eine Property Ganzer Name, welche den Ganzen Namen einer Property Zurückgibt. Über die Property Ganzer Name darf kein Wert gesetzt werden.

Hinweis: Nutze eine Property/Eigenschaft. Sie verhalten sich wie Variablen, sind jedoch sicherer und flexibler, da sie mit Logik ausgestattet werden können, um Werte zu prüfen oder zusätzliche Aktionen auszuführen.

Allgemein:

```
// In einem Struct oder einer Klasse
public TYP Prop
{
    get { // get ist optional
        // Mache irgendetwas
        return someValue;
    }
    set { // set ist optional
        // Bekommt unsichtbaren Parameter value
        // Mache irgendetwas Meistens veraendern von Variablen
    }
}

// In Main Methode
```

```
Console.WriteLine(meinStruct.Prop) // ruft get von Prop auf  
meinStruct.Prop = 42; // ruft set von Prop mit value=42 auf
```

Beispiel: Wir haben eine Thermometer Klasse

```
struct Thermometer{
    public double celsius;
    public double Fahrenheit
    {
        get { return (celsius * 9/5)+32; } // Celsius -> Fahrenheit
        set { celsius = (value - 32)*5/9; } // Fahrenheit -> Celsius
    }
}
// In Main Methode
var thermometer = new Thermometer();
thermometer.celsius = 30;
Console.WriteLine(thermometer.Fahrenheit); // 86
thermometer.Fahrenheit = 77
Console.WriteLine(thermometer.celsius); // 25
```

g) GanzerName Property mit set:

Füge zu der GanzenNamen Property eine set Methode bzw. einen set Zugriff hinzu.

Hinweis: Die Methode .Split(seperator) teilt einen Text in eine string Liste auf. Z.B:

```
var datum = "26.11.2024";

string[] datumsTeile = datum.Split(".");
string tag = datumsTeile[0]; // 26
string monat = datumsTeile[1]; // 11
string jahr = datumsTeile[2]; // 2024
```

Aufgabe 3: Dateien bearbeiten: Lesen, Zählen, Schreiben

Hinweis: In C# kann eine Datei mit der `StreamReader`- oder `StreamWriter`-Klasse verarbeitet werden. Nach dem Gebrauch ist es wichtig, die Datei zu schließen, um Ressourcen freizugeben.

Dateien lesen

Zeichen für Zeichen:

```
var filePath = "sample.txt";
var reader = new StreamReader(filePath);
int character;
while ((character = reader.Read()) != -1)
{
    // Mache etwas!
}
reader.Close();
```

Zeile für Zeile:

```
var filePath = "sample.txt";
var reader = new StreamReader(filePath);
string line;
while ((line = reader.ReadLine()) != null)
{
    // Mache etwas!
}
reader.Close();
```

Nutze für folgende Aufgaben diese Beispiel Log Datei

```
# Activity Log
Logged In
Error While User Logged In: Invalid Password
Logged In
Logged In
```

- Dateiinhalte auf der Konsole ausgeben. Erstelle eine Methode, die den gesamten Inhalt einer Datei Zeile für Zeile auf der Konsole ausgibt.
- Anzahl von Buchstaben L in einer Datei zählen. Erstelle eine Methode, die alle Buchstaben L in einer Datei zählt.
- Anzahl von `Logged In` - Zeilen in einer Datei zählen. Erstelle eine Methode, die die Anzahl der Zeilen, welche exakt `Logged In` sind, zählt.
- Eine Datei mit Zahlen befüllen. Erstelle eine Methode, die eine Datei `numbers.txt` erstellt und die Zahlen von 0 bis 1000 in die Datei schreibt.

Hinweis: Eine Datei erstellen, in eine Datei schreiben:

```
var filePath = "datei.txt";
var writer = new StreamWriter(filePath);
```

```
// Mache etwas: Write, WriteLine wie bei Console.Write
writer.Write("")
writer.WriteLine("")
writer.Close();
```

Aufgabe 4: 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) Schreibe einen Konstruktor und ToString (schwer)
Überlege dir einen sinnvollen Konstruktor und eine ToString Methode für das Handy.

Aufgabe 5: Erstelle eine Anwesenheitsliste

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

Hinweis: Nutze folgendes Person struct

```
struct Person
{
    public string Name;

    public Person(string name)
    {
        Name = name;
    }

    public override string ToString()
    {
        return Name;
    }
}
```

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