



Software Development

Quellenangaben:

- Jürgen Kotz, München, Visual C# 2019, Carl Hanser Verlag München
- Michael Bonacina, C# Programmieren für Einsteiger (2. Aufl.), BMU Verlag
- <https://docs.microsoft.com/de-de/dotnet/csharp/programming-guide>

Datenstrukturen

Datenstrukturen

- Arrays
- Mehrdimensionale Arrays
- Listen für identische Datentypen
- Listen für unterschiedliche Datentypen

Arrays

- Speichern von mehreren Variablen des gleichen Typs in eine Datenstruktur
- Array kann eindimensional, mehrdimensional oder verzweigt sein
- Anzahl Dimensionen und Länge der Dimensionen bei der Instanziierung festgelegt. Werte können während der Lebensdauer der Instanz nicht geändert werden
- Arrays sind nullbasiert (Index beginnt bei 0 und endet bei n-1)
- Arrayelemente können beliebigen Typ aufweisen
- foreach-Iteration kann verwendet werden

Arrays

Deklaration eines einfachen Array

C #

```
int[] myArray = new int[5];  
myArray[1] = 5;  
Console.WriteLine (myArray[1]);
```

	5			
--	---	--	--	--

Deklaration eines einfachen Array mit Werten

C #

```
int[] myArray = new int[] { 1, 3, 5, 7, 9 };  
Console.WriteLine (myArray[1]);
```

1	3	5	7	9
---	---	---	---	---

Alternative Schreibweise

C #

```
int[] myArray = { 1, 3, 5, 7, 9 };  
Console.WriteLine (myArray[0]);
```

Arrays

Deklaration eines mehrfachen Array

C #

```
int[,] myArray = new int[2, 3]; // 2 Zeile, 3 Spalten  
myArray[0, 1] = 5; // Index beginnt bei 0  
Console.WriteLine (myArray[0, 1]);
```

	5	

Deklaration eines mehrfachen Array mit Werten

C #

```
int[,] myArray = { {1, 2, 3}, {4, 5, 6} }; // 3 Zeilen  
                                              // 2 Spalten  
Console.WriteLine (myArray[0, 1]);
```

1	2	3
4	5	6

Übungen

Uebungen\Kurseinheit 4\Übung Array.pdf

Uebungen\Kurseinheit 4\Übung König und
seine Gefangenen.pdf

Listen für variable Anzahl Werten

- keine feste Länge
- Grösse passt sich dynamisch an
- Unterscheidung identischer/unterschiedlicher Datentyp

Listen für identische Datentypen

- Nur Daten des gleichen Typs

Deklaration einer Liste vom Typ int

C #

```
List<int> meineListe = new List<int>();
```

Hinzufügen von Werten

C #

```
meineListe.Add(12);
```

Entfernen von Werten

C #

```
meineListe.Remove(12);  
// oder  
meineListe.RemoveAt(0);
```



Listen für unterschiedliche Datentypen

- Daten unterschiedlichen Typs

Deklaration einer ArrayList

C #

```
ArrayList meineListe = new ArrayList();
```

Hinzufügen von Werten

C #

```
meineListe.Add("Hallo");  
meineListe.Add(25);
```

Entfernen von Werten

C #

```
meineListe.Remove("Hallo");  
// oder  
meineListe.RemoveAt(0);
```




Übungen

Uebungsaufgaben 5.6 s.60