

Generische Listen

generische Datentypen

- durch generische Datentypen kann der Datentyp von Mitgliedern (Methoden/Eigenschaften) variabel gestaltet werden
- generische Typen werden mit dem Platzhalter „T“ festgelegt
- innerhalb der Klasse können dann Objekte des Datentyps T erzeugt werden
- oft für Auflistungen benutzt, welche egal bei welchem Datentyp das gleiche Grundverhalten aufweisen sollen

List<T>

- eine Liste von Objekten <T>, auf die per Index zugegriffen werden kann
- vordefinierte Methoden zum Hinzufügen, Entfernen, Durchsuchen, Sortieren und Bearbeiten

```
List<string> stringListe = new List<string>();
stringListe.Add("Das");
stringListe.Add("ist");
stringListe.Add("eine");
stringListe.Add("Liste");
stringListe.Add("mit");
stringListe.Add("strings");
foreach (var item in stringListe)
{
    Console.WriteLine(item);
}

//Das
//ist
//eine
//Liste
//mit
//strings
```

List<T> - nützliche Funktionen

Funktionsname	Beschreibung
Add(T)	fügt der Liste ein Objekt hinzu
Remove(T)	entfernt das erste Objekt T
Clear()	entfernt alle Elemente aus der Liste
Find(Predicate<T>)	gibt ein Objekt zurück, welches dem Predicate<T> entspricht
Contains(T)	gibt zurück ob das Objekt T in der Liste enthalten ist
Exist(Predicate<T>)	gibt zurück ob ein Objekt, welches dem Predicate<T> entspricht, existiert
Sort(Comparison<T>)	sortiert die Liste mithilfe der angegebenen Comparison<T>

- Die komplexeren Methoden (Find(), Contains(), etc.) finden sich als Erweiterungsmethoden im Namespace `System.Linq`

spezielle Listen – Dictionary<TKey, TValue>

- Entspricht ‚Tabelle mit 2 Spalten‘
- Ordnet jeweils einem Key (TKey) ein Value (TValue) zu
- Add(TKey, TValue): Element hinzufügen
- über Key-Angabe Zugriff auf den Value
- ContainsKey(TKey) prüft, ob ein Key vorhanden ist
- ContainsValue(TValue) prüft, ob ein Value vorhanden ist

```
Dictionary<int, string> stringDictionary = new Dictionary<int, string>();  
stringDictionary.Add(5, "Element 1");  
stringDictionary.Add(10, "Element 2");  
string value10 = stringDictionary[10]; // "Element 2"  
stringDictionary.ContainsKey(5); //true  
stringDictionary.ContainsValue("Element 3"); //false
```