

# Indexer

Ingo Köster

Diplom-Informatiker (FH)

# Indexer

---

- › Zugriff auf die Elemente einer Klasse oder Struktur
- › Die Klasse String verwendet den sog. Indexer
- › `string s = "Hallo";`
- › `Console.Write(s[0]);`    `// 'H'`
- › `Console.Write(s[2]);`    `// 'l'`

# Eigene Indexer erstellen

---

- › Entscheidend ist das `this[...]` innerhalb der Klasse

```
class Satz
{
    string[] wörter = "Dies ist ein Satz".Split();

    // Der Indexer
    public string this[int wortNummer]
    {
        get { return wörter[wordNummer]; }
        set { wörter[wordNummer] = value; }
    }
}
```

# Indexer

---

› Verwendung des Indexers

```
Satz einSatz = new Satz();  
Console.WriteLine(einSatz[1]); // ist  
einSatz[2] = "kein";
```

# Indexer

---

- › Indexer können verschiedene Datentypen verwenden (z.B. string)

```
class IndexedNames
{
    string[] names = new string[5];

    public int this[string name]
    {
        get { ... }
    }
}
```

# Indexer

---

- › Es können mehrere Indexer pro Typ verwendet werden
  - › Parameter müssen sich im Datentyp unterscheiden
- › Indexer können überladen werden
- › Ein Indexer kann mehr als einen Parameter verwenden

```
public string this [int arg1, string arg2]
{
    get {...}
    set {...}
}
```

# Indexer

---

- › Können Generics verwenden

```
public T this[int i]
{
    get { ... }
}
```

- › Indexer in Schnittstellen

```
interface ISomeInterface
{
    string this[int index] { get; set; }
}
```