

Generisches Delegate

Ingo Köster

Diplom-Informatiker (FH)

Generische Delegaten

› Ein generisches Delegat anlegen:

```
public delegate void MyDelegate<T>(T obj);
```

› Generisches Delegat instanziiieren:

```
MyDelegate<int> del = MyEventHandler;
```

Beispiele – Deklaration

```
// Außerhalb der Klasse
```

```
delegate void AdditionsHandler<T>(T a, T b);
```

```
class MyClass<T>
```

```
{
```

```
    // Variable innerhalb der Klasse
```

```
    public AdditionsHandler<T> additionsDelegate;
```

```
}
```

Beispiele – Instanziierung mit int

```
MyClass<int> my = new MyClass<int>();
```

```
my.additionsDelegate = AddiereInt;
```

```
my.additionsDelegate(1, 5);
```

```
...
```

```
static void AddiereInt(int x, int y)
{
    Console.WriteLine(x + y);
}
```

Beispiele – Instanziierung mit double

```
MyClass<double> my = new MyClass<double>();
```

```
my.additionsDelegate = AddiereDouble;
```

```
my.additionsDelegate(3.14, 5.0);
```

```
...
```

```
static void AddiereDouble(double x, double y)
{
    Console.WriteLine(x + y);
}
```

Mit generischem Rückgabewert

```
delegate T MeinHandler<T>(T parameter);
```

```
class MyClass<T> {  
    public MeinHandler<T> meinDelegate;  
}
```

```
MyClass<string> my = new MyClass<string>();  
my.meinDelegate = Up;
```

```
Console.WriteLine(my.testDelegate("xyz")); // XYZ
```

```
static string Up(string x) {  
    return x.ToUpper();  
}
```

Generisches Delegat mit verschiedenen Typen

```
delegate string MeinHandler<T1, T2>(T1 p1, T2 p2);
```

```
class Test<T1, T2> {  
    public MeinHandler<T1, T2> meinDelegate;  
}
```

```
Test<string, int> t = new Test<string, int>();  
t.meinDelegate = Ausgabe;  
Console.WriteLine(t.meinDelegate("Hallo: ", 10));
```

```
static string Ausgabe(string a, int b) {  
    return a + b.ToString();  
}
```

Generische Delegaten einschränken

- › Generische Delegaten durch Constraints einschränken

- › Schlüsselwort `where`

- › Bei der Deklaration

```
delegate void MyDelegateHandler<T>(T obj) where T : ClassA,  
IMyInterface;
```