

# Delegaten und Events

Delegaten sind in C# aufgebaut wie normale Methoden, jedoch besitzen sie keinen Code der ausgeführt wird, sondern weisen lediglich auf eine Methode mit Code hin.

Das bedeutet, dass die Delegaten genau so wie Methoden

einen Zugriffsmodifikator (private, public ), einen Rückgabewert und Parameter haben. Die Parameter müssen aber in der Anzahl und Datentyp den Methoden identisch sein.

[crayon-6767a0d969219152239189/]

Interessant werden Delegaten in Verbindung mit Events auf Deutsch Ereignisse. Events werden ausgelöst. Wenn man einen Button klickt, löst man das Click-Event aus. In WPF und Windows Forms haben die Steuerelemente z.B. viele vordefinierte Events. So kann man ein Mouseover Event auslösen lassen, wenn man z.B. mit der Maus über ein Steuerelement fährt. Sobald nun ein Event ausgelöst wird, versucht das an ihm hängende Delegat eine Methode auszulösen mit den Rückgabewerten und Parameter, die dem Delegat zur Verfügung stehen. Das Interessante ist, dass ein Delegat unabhängig vom Zugriffsmodifikator arbeitet. An beide wird eine Methode durch Überladung angehängt " += " oder abgezogen " -= „.

Mal ein kleines Beispiel, wie wir mithilfe eines Delegaten und Event von einem neuen Window in das Mainwindow Label etwas schreiben können:

MainWindow:

[crayon-6767a0d96921e668174066/]

Das neue Window1:

[crayon-6767a0d969220966398582/]

Hier sehen wir, dass zunächst im MainWindow dem Event vom

Objekt w1 die Methode wnd\_MyCustomEvent hinzugetan wurde. Jedesmal also, wenn das Event ausgeführt wird, wird auch diese Methode ausgeführt.

Im Window1 wird nun die Methode durch das klicken auf den cmd\_w1\_Button das Event MyEvent ausgelöst und dem dazugehörigen Delegaten wird der Parameter von 100 übergeben.

Abschließend wird nun im MainWindow dieser Wert angezeigt

Der nächste Schritt, den man gehen könnte ist, noch ein neues Window zu erstellen und dort eine Methode hinzufügen:

```
[crayon-6767a0d969221182395920/]
```

Dem MainWindow fügen wir hinzu:

```
[crayon-6767a0d969222445791287/]
```

Ausgabe ist eine MessageBox mit dem Wert 100 aus der Klasse in Window2 heraus