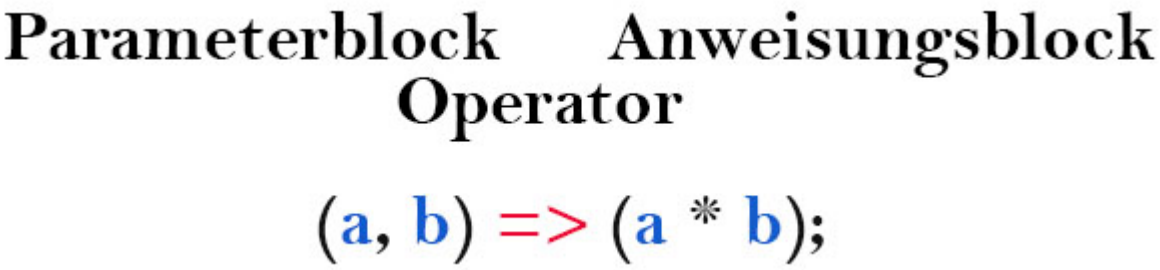
Lambda Ausdrücke oder anonyme Methoden

Lambda Ausdrücke werden in Kombination mit Delegaten dazu verwendet um schnelle Anweisungen auszuführen und sich dabei jede Menge Code zu ersparen.

Ohne Lambda und anonymer Methode für ein Delegatenaufruf in etwa so aussehen:



Neue Eigenschaft vom Typ delegate erstellen. (Rückgabewert ist ein int, mit 2 in Parameter a und b)
[crayon-662b8f6c4ee42234874429/]
Dann müssen wir eine Methode haben, die ebenfalls denselben Rückgabewert und Parameter besitzt (Anzahl und Typ sind entscheidend)
[crayon-662b8f6c4ee49447258866/]
Nun könnte man ein Objekt vom den oben erstellten MyDelegate erstellen und dem die Methode addiere zuweisen:
[crayon-662b8f6c4ee4a048839962/]
Durch den Aufruf des Objektes dlg zeigt der Delegat auf die Methode addiere, führt diese aus und gibt den Rückgabewert zurück:
[crayon-662b8f6c4ee4c315412669/]

Dies ist in der Tat etwas umständlich. Vor allem benötigt man immer ein Methode, die man dem Delegaten immer zuweisen muss.

schneller geht es mit den Lambda Ausdrücken.:
[crayon-662b8f6c4ee4d755089551/]
Die delegaten delegat1 bis delegat 5 tun immer dasselbe, nur wird der Code dadurch sehr kompakter und wenn man delegaten erstmal verstanden hat, wird es auch übersichtlicher.

Hat man einen Delegaten ohne Rückgabewert, würde der Lambdaausdruck mit Klammer auf- Klammer zu sein:
[crayon-662b8f6c4ee4e423392082/]

Wegen den => Operator werden die Lampdaausdrücke oft mit Linq verwechselt. Das eine hat mit dem anderen aber nichts zu tun