

Virtuelle und dynamische Methoden

[Themengruppen](#)

[Siehe auch](#)

Mit Hilfe der Direktiven **virtual** und **dynamic** können Methoden als virtuell oder dynamisch deklariert werden. Virtuelle und dynamische Methoden können im Gegensatz zu statischen Methoden in abgeleiteten Klassen überschrieben werden. Beim Aufrufen einer überschriebenen Methode bestimmt nicht der deklarierte, sondern der aktuelle Typ (also der Typ zur Laufzeit) der im Aufruf verwendeten Klassen- bzw. Objektvariable, welche Implementierung aktiviert wird.

Um eine Methode zu überschreiben, braucht sie nur mit der Direktiven **override** erneut deklariert zu werden. Dabei müssen Reihenfolge und Typ der Parameter sowie der Typ des Rückgabewertes (falls vorhanden) mit der Deklaration in der Vorfahrklasse übereinstimmen.

Im folgenden Beispiel wird die in der Klasse TFigure deklarierte Methode Draw in zwei abgeleiteten Klassen überschrieben:

```
type
  TFigure = class
    procedure Draw; virtual;
  end;
  TRectangle = class(TFigure)
    procedure Draw; override;
  end;
  TEllipse = class(TFigure)
    procedure Draw; override;
  end;
```

Ausgehend von diesen Deklarationen zeigt der folgende Programmcode, wie sich der Aufruf einer virtuellen Methode durch eine Variable auswirkt, deren aktueller Typ zur Laufzeit geändert wird.

```
var
  Figure: TFigure;
begin
  Figure := TRectangle.Create;
  Figure.Draw; // Ruft TRectangle.Draw auf.
  Figure.Destroy;
  Figure := TEllipse.Create;
  Figure.Draw; // Ruft TEllipse.Draw auf.
  Figure.Destroy;
end;
```

Nur virtuelle und dynamische Methoden können überschrieben werden. Alle Methoden können jedoch überladen werden (siehe [Methoden überladen](#)).

Unterschiede zwischen virtuellen und dynamischen Methoden

Virtuelle und dynamische Methoden sind von der Semantik her identisch. Sie unterscheiden sich nur bei der Implementierung der Aufrufverteilung zur Laufzeit. Virtuelle Methoden werden auf Geschwindigkeit, dynamische Methoden auf Code-Größe optimiert.

Im allgemeinen kann mit virtuellen Methoden polymorphes Verhalten am effizientesten implementiert werden. Dynamische Methoden sind hilfreich, wenn in einer Basisklasse eine große Anzahl überschreibbarer Methoden deklariert ist, die von vielen abgeleiteten Klassen geerbt, aber nur selten überschrieben werden.

[Unterschiede zwischen Überschreiben und Verdecken](#)

[reintroduce](#)

[Abstrakte Methoden](#)

Abstrakte Methoden

[Themengruppen](#)

[Siehe auch](#)

Eine abstrakte Methode ist eine virtuelle oder dynamische Methode, die nicht in der Klasse implementiert wird, in der sie deklariert ist. Die Implementierung wird erst später in einer abgeleiteten Klasse durchgeführt. Bei der Deklaration abstrakter Methoden muß wie im folgenden Beispiel die Anweisung **abstract** nach **virtual** oder **dynamic** angegeben werden:

```
procedure DoSomething; virtual; abstract;
```

Eine abstrakte Methode kann nur in einer Klasse (bzw. Instanz einer Klasse) aufgerufen werden, in der sie überschrieben wurde.