

Sichtbarkeit von Klassenelementen

[Themengruppen](#)

[Siehe auch](#)

In einer Klasse hat jedes Element ein Sichtbarkeitsattribut, das durch die reservierten Wörter **private**, **protected**, **public**, **published** und **automated** angegeben wird. Im folgenden Beispiel wird die Eigenschaft *Color* als **published** deklariert:

```
published property Color: TColor read GetColor write SetColor;
```

Die Sichtbarkeit bestimmt, wo und wie auf ein Element zugegriffen werden kann. **private** entspricht der geringsten, **protected** einer mittleren und **public**, **published** und **automated** der größten Sichtbarkeit.

[private-, protected- und public-Elemente](#)

[published-Elemente](#)

[automated-Elemente](#)

Ein Element ohne Attribut erhält automatisch die Sichtbarkeit des vorhergehenden Elements in der Deklaration. Die Elemente am Anfang einer Klassendeklaration ohne explizite Sichtbarkeitsangabe werden standardmäßig als **published** deklariert, wenn die Klasse im Status **{ $\$M+$ }** compiliert oder von einer mit **{ $\$M+$ }** compilierten Klasse abgeleitet wurde. Andernfalls erhalten sie das Attribut **public**.

Aus Gründen der Lesbarkeit sollten Sie die Elemente einer Klassendeklaration nach ihrer Sichtbarkeit gruppieren. Nehmen Sie daher zuerst alle **private**-Elemente, anschließend alle **protected**-Elemente usw. in die Deklaration auf. Bei dieser Vorgehensweise braucht das Sichtbarkeitsattribut nur einmal angegeben zu werden, und es markiert immer den Anfang eines neuen Deklarationsabschnitts. Eine typische Klassendeklaration sieht somit folgendermaßen aus:

```
type  
  TMyClass = class(TControl)  
    private  
    ... { private-Deklarationen }  
    protected  
    ... { protected-Deklarationen }  
    public  
    ... { public-Deklarationen }  
    published  
    ... { published-Deklarationen }  
  end;
```

Sie können die Sichtbarkeit eines Elements in einer untergeordneten Klasse durch Redeklarieren erhöhen, jedoch nicht verringern. So kann beispielsweise eine **protected**-Eigenschaft in einer abgeleiteten Klasse als **public** deklariert werden, nicht aber als **private**. Außerdem können **published**-Elemente nicht zu **public**-Elementen gemacht werden. Weitere Informationen zu diesem Thema finden Sie im Abschnitt [Eigenschaften überschreiben und neu deklarieren](#).

private-, protected- und public-Elemente

[Themengruppen](#)

[Siehe auch](#)

Auf ein **private**-Element kann nur innerhalb des Moduls (Unit oder Programm) zugegriffen werden, in dem die Klasse deklariert ist. Mit anderen Worten: eine **private**-Methode kann nicht von anderen Modulen aufgerufen werden, und als **private** deklarierte Felder oder Eigenschaften können nicht von anderen Modulen gelesen oder geschrieben werden. Indem Sie verwandte Klassendeklarationen im selben Modul zusammenfassen, können Sie diesen Klassen also den Zugriff auf alle **private**-Elemente ermöglichen, ohne die Elemente anderen Modulen bekanntzumachen.

Ein **protected**-Element ist innerhalb des Moduls mit der Klassendeklaration und in allen abgeleiteten Klassen (unabhängig davon, in welchem Modul sie deklariert sind) sichtbar. Mit anderen Worten: auf ein **protected**-Element können alle Methoden einer Klasse zugreifen, die von der Klasse mit der Elementdeklaration abgeleitet ist. Mit diesem Sichtbarkeitsattribut werden also Elemente deklariert, die nur in den Implementierungen abgeleiteter Klassen verwendet werden sollen.

Ein **public**-Element unterliegt keinerlei Zugriffsbeschränkungen. Es ist überall dort sichtbar, wo auf seine Klasse verwiesen werden kann.