**Abstrakte Klassen, Methoden und Properties, versiegelte Klassen**

**Abstrakte Klassen**

Klassen, die lediglich ihr „Erbmaterial“ an andere Klassen weitergeben und selbst nicht instanziiert werden dürfen, bezeichnet man als **abstrakt**.  
Um also zu verhindern, dass von einer bestimmten Klasse Instanzen gebildet werden, müssen Sie diese als **abstract** markieren.

**Beispiel**

internal **abstract** class ElektroGeraet

{

internal string Marke { get; set; }

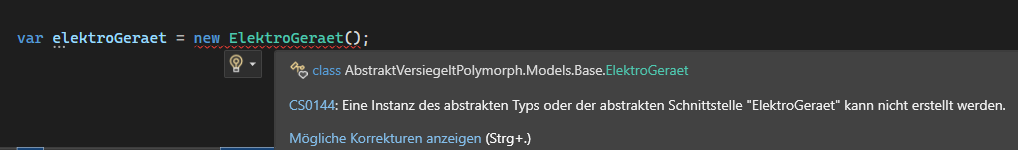
internal string Modell { get; set; }

public int LeistungInWatt { get; set; }

}

**Logische Schlussfolgerung:** Die Definition eines Modells namens **ElektroGeraet** ist zu allgemein und zu unzureichend, um daraus **sinnvolle** Objekte modellieren zu können. Man kann sich unter ElektroGeraetrecht wenig vorstellen, es ist– umgangssprachlich ausgedrückt „zu abstrakt“.

Vorkompilierungsfehler bei Versuch abstrakten Typ ElektroGeraet zu instanziieren

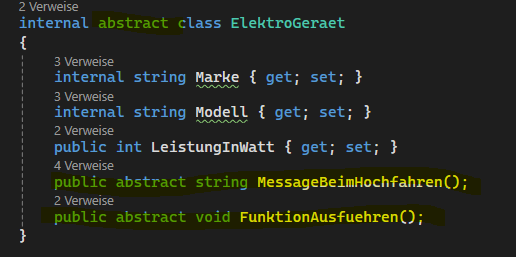


**Abstrakte Methoden und Properties**

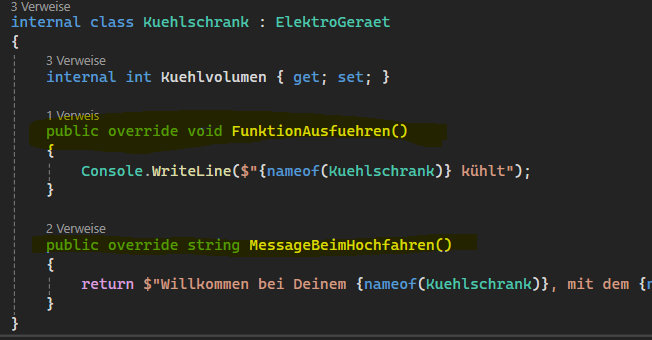
Finden grundsätzlich Anwendung in abstrakten Klassen. Sie werden mit dem Zusatz **abstract** modifiziert, enthalten selbst keinen Code und **müssen** in den abgeleiteten Klassen mit Modifizierer **override** überschrieben werden.

**Beispiel**

Definition in Elternklasse



Überschreibung / Implementierung in Kindklasse



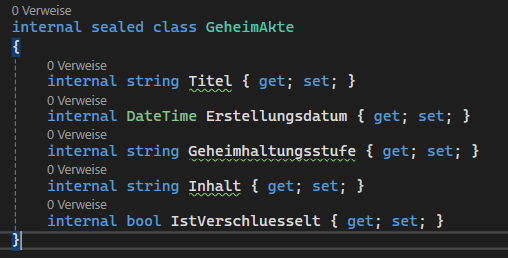
**Übung zu Abstraktion**

Erstellen Sie eine abstrakte Klasse **Kaffee**, von der andere Klassen (wie Melange, usw.) erben. Versehen Sie die Elternklasse mit einer **abstrakten** Methode **Geschmack**. Diese soll in den abgeleiteten Klassen den Charakter der jeweiligen Kaffeesorte ausdrücken können.

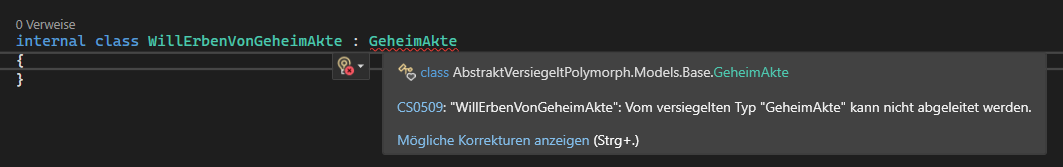
**Versiegelte Klassen**

Wenn Sie eine bestimmte Klasse unbedingt vor Ableitung durch andere Klassen **schützen** möchten, können Sie diese mit dem Schlüsselwort **sealed** versehen.

Klasse GeheimAkte ist **versiegelt** und darf somit kein Nachkommen haben.



Vorkompilierungsfehler bei Versuch von einer versiegelten Klasse zu erben



**Übung**

Erstellen Sie eine versiegelte Klasse **PrivatKunde** mit ein paar sinnvollen Eigenschaften und versuchen Sie eine zu erstellende Klasse namens **WillAuchPrivatKundeSein** von dieser abzuleiten.