



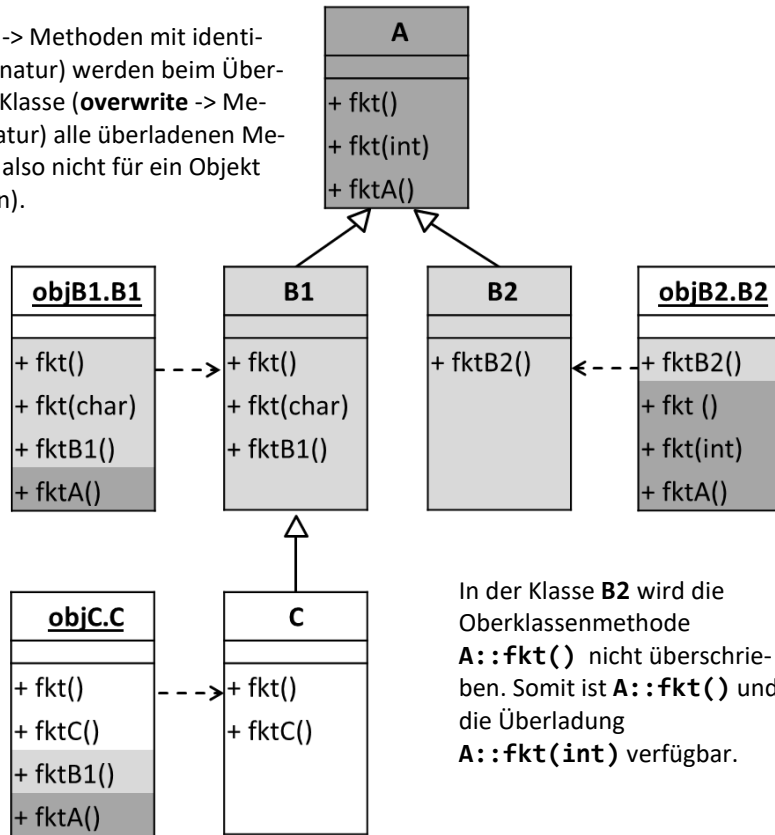
Überschreiben ist ein Vorgang, bei dem in der abgeleiteten Klasse Methoden mit der identischen Signatur einer Basisklassenmethode deklariert werden.

Im Gegensatz zum Überladen (**overload** -> Methoden mit identischem Namen und unterschiedlicher Signatur) werden beim Überschreiben einer Methode der erbenen Klasse (**overwrite** -> Methode mit identischem Namen und Signatur) alle überladenen Methoden der Basis-Klasse überdeckt (sind also nicht für ein Objekt der abgeleiteten Klasse direkt vorhanden).

Die Methode **B1::fkt()** überschreibt **A::fkt()**. Dadurch sind auch alle Überladungen (**A::fkt(int)**) überdeckt.

In der Klasse **C** wird die Oberklassenmethode **B::fkt()** und somit auch **A::fkt()** überschrieben. Damit sind alle Überladungen (**A::fkt(int)** und **B1::fkt(char)**) überdeckt und nicht direkt aufrufbar.

In der Klasse **B2** wird die Oberklassenmethode **A::fkt()** nicht überschrieben. Somit ist **A::fkt()** und die Überladung **A::fkt(int)** verfügbar.



Um trotzdem die überdeckten Methoden der Oberklasse aufzurufen, wird die Methode mit Klassenname und Bereichsoperator angesprochen (also **objC.A::fkt(int)**; bzw. **objC.B1::fkt(char)**);).

In dem oben beschriebenen Beispiel wird beim Kompilieren festgelegt, welche Methode zu welchem Objekt „gebunden“ wird. Man spricht vom statischen Binden. Soll erst zur Laufzeit die passende Methode gewählt werden (dynamisches Binden), muss die Methode als **virtual** deklariert werden. In diesem Fall spricht man von **polymorphen Verhalten** bzw. **Polymorphie**.

Da ein Objekt der Klasse B1 immer auch ein Objekt vom Typ der Oberklasse ist (siehe Vererbung), kann eine Referenzvariable vom Typ A auf ein Objekt vom Typ B1 verweisen.

