LU11a - Abstrakte Basisklassen

Eine abstrakte Klasse kann als Blaupause für andere Klassen betrachtet werden. Sie ermöglicht es, eine Reihe von Methoden zu erstellen, die **in allen untergeordneten Klassen implementiert werden müssen** (= Vertrag für die Implememtation).

Was ist eine abstrakte Klasse

Eine Klasse, die eine oder mehrere abstrakte Methoden enthält, wird als abstrakte Klasse bezeichnet. Eine abstrakte Methode ist eine Methode, die eine Deklaration, aber keine Implementierung hat.

Wenn eine gemeinsame Schnittstelle für verschiedene Implementierungen einer Komponente bereitgestellt werden soll, verwenden wir eine abstrakte Klasse.

In der UML wird eine abstrakte Klasse durch `{abstract}` markiert, während eine abstrakte Methode *kursiv* angeschrieben wird.

Beispiel: Übergeordneter Begriff ''Medium'' als Träger einer Niederschrift

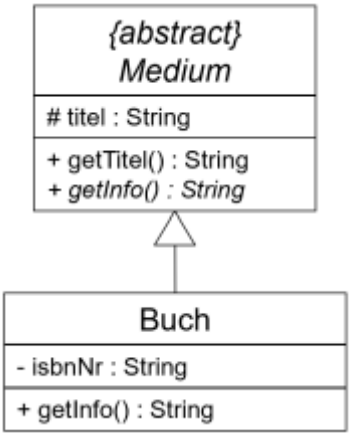


Abb: UML-Diagramm mit abstrakter Klasse

Die Klasse `Medium` kann nicht instanziert werden, da sie als `abstract` deklariert ist. Es bedingt immer eine konkrete Ableitung (hier `Buch`) der Klasse. Die Methode `getInfo` wird als abstrakte Methode deklariert und muss in der abgeleiteten Klasse konkretisiert werden.

Wozu abstrakte Basisklassen verwenden?

Indem eine abstrakte Basisklasse definiert wird, können gemeinsame Anwendungsprogrammschnittstelle (API) für eine Reihe von Unterklassen definiert werden. Diese Funktion ist besonders nützlich in Situationen, in denen ein Drittanbieter Implementierungen

bereitstellt, z.B. mit Plugins. Sie kann aber auch helfen, wenn in einem grossen Team oder mit einer grossen Codebasis gearbeitet wird. Eine Situation in der es schwierig oder gar nicht möglich ist, alle Klassen im Kopf zu behalten.

© Danie Fahrni, René Probst

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/de/modul/m320_2024/learningunits/lu11/abstrakte_basisklasse

Last update: **2026/08/30 14:06**