

# LU05c - Generische Typisierung

In Java können Klassen und auch Interfaces einen oder gar mehrere generische Typen haben. Beispielsweise hat die Klasse `java.util.ArrayList<E>` als auch das implementierte Interface `java.util.List<E>` jeweils den generischen Typ „E“. Bei der Klasse `java.util.HashMap<K, V>` und dem Interface `Map<K, V>` sind es die beiden generischen Typen „K“ und „V“.

Generische Typisierung ist praktisch, um den Entwickler bzw. die Entwicklerin zu leiten und die Absicht zur Verwendung der Klasse respektive des Interfaces darzulegen.

Der generische Typ wird bei einer Instanzierung mitgegeben:

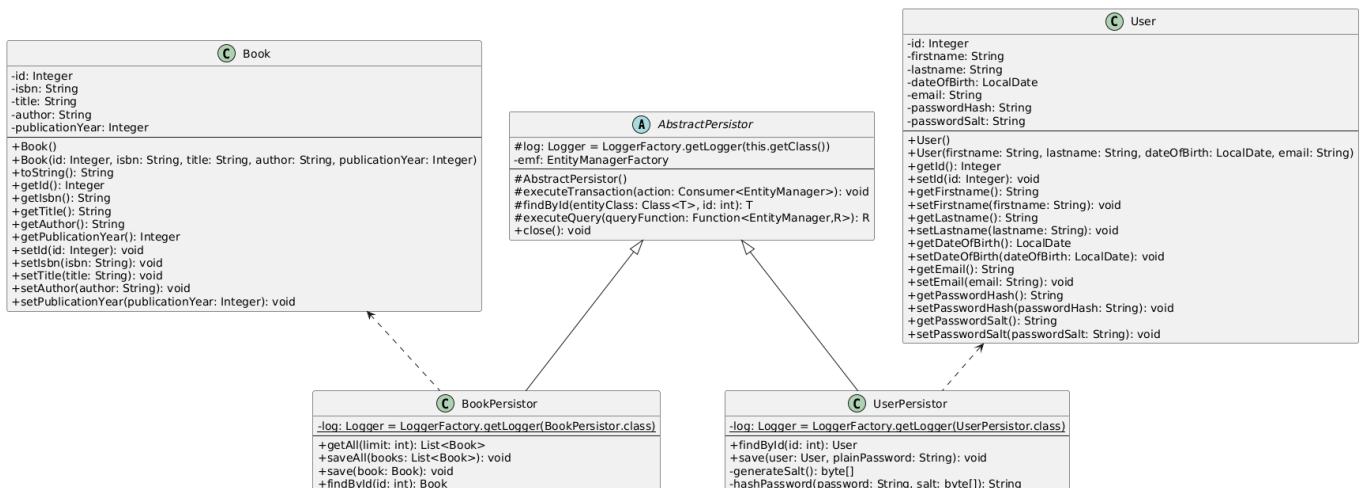
```
var myList = new ArrayList<String>();
```

Der Diamond-Operator `<>` kann bei der instanziierten Klasse benutzt werden, wenn der generische Typ aus dem Klasse bzw. dem Interface abgeleitet werden kann.

```
List<String> myList = new ArrayList<>();
```

Eine Klasse oder ein Interface kann auch einen konkreten generischen Typ fix vererbt bekommen. In diesem Fall wird bei der Verwendung von `BookPersistor` immer `Book` als generischen Typ verwendet.

```
public class BookPersistor extends AbstractPersistor<Book> { ... }
```



From:

<https://wiki.bzz.ch/> - BZZ - Modulwiki

Permanent link:

<https://wiki.bzz.ch/de/modul/ffit/3-jahr/java/learningunits/lu05/c?rev=1758000244>

Last update: 2025/09/16 07:24

