

II.4.6 Collections

Mittwoch, 21. Dezember 2016 09:00

Collections: generische
verdefinierte Datentypen
für Sammlungen von
Objekten

Bsp. für Collection-Klassen:

- **LinkedList**: verkettete Liste
Zugriff am Anfang + Ende der
Liste schnell, wird zur
Mitte hin langsamer.
Schnelles Einfügen neuer Werte.
- **ArrayList**: als Array reali-
sierte Liste, schneller Zu-
griff. Wenn Größe nicht
reicht, wird in neues Array
unkopiert.
- **Sets (Mengen)** haben keine Du-
plikate
 - * **HashSet**: Gleich schnelles

einfügen, löschen, suchen

* TreeSet: Elemente als
Baum nach Größe gespeichert,
logarithm. Aufwand für
suchen, löschen

• Maps: Schlüssel-Wert Paare

Iterator $\hat{=}$ Zeiger auf ein
Element einer Sammlung

Für konkrete Collection-Klassen
wie z.B. LinkedList (die
Iterator implementiert), liefert
iterator() einen Iterator
einer konkreten Klasse

LinkedListIterator zurück
(die Iterator implementiert).

Iterator ist "verbraucht", wenn
er am Ende der Sammlung an-

gekommen ist.

Für jede Klasse, die `Iterable` implementiert, kann man die `for-each` Schleife benutzen, um mit Hilfe des Iterators die Werte der Sammlung zu durchlaufen.

Bsp: Collections in Kombination mit Autoboxing + Unboxing

Wenn über eine Sammlung mit einem Iterator iteriert wird, dann darf die Sammlung während der Iteration nicht durch neue Elemente oder Löschen verändert werden (außer durch die `remove`-Methode des Iterators).

Sonst: ConcurrentModification

Exception