II.4.6 Collections

Montag, 3. Dezember 2018 11:30

Vordef. Collection-Klassen:

- LinkedList: verkettete Liste
  Zugriff an den beiden Enden der Liste schnell, wird zur Mitte hin langsamer
- ArrayList: als Array realisierte Liste.
  Schneller Zugriff auf alle Elemente, aber wenn Array-Größe nicht ausreicht, dann werden alle Werte in ein größeres Array kopiert
- Set: Sammlung ohne Duplikate (HashSet, TreeSet, ...)
- Maps: sind keine Unterklassen von Collection, werden aber auch zum Collection-Framework gezählt (Schlüssel-Wert Paare).

Iterator ≙ Zeiger auf ein Element der Sammlung

Ein Iterator hat 3 Methoden:

hasNext: sagt, ob es noch einen weiteren Wert in der Sammlung gibt

next: liefert den nächsten Wert u. setzt Iterator weiter

remove: löscht den Wert, über den der Iterator zuletzt gelaufen ist.

Wenn Iterator am Ende der Sammlung ist, ist er "verbraucht".

Iterator für LinkedList wird durch Konkrete Klasse imple-

mentiert, in der hasNext, next, remove geeignet überschrieben werden. Wird typischerweise mit current + previous-Zeiger realisiert.

foreach-Schleife kann nicht nur für Arrays benutzt werden, sondern auch für Collections (und für eigene Klassen, die Collection-Interface implementieren).

Primitive Datentypen können eigentlich nicht in LinkedList<T> gespeichert werden, denn T kann nur mit Klassen- oder Interface-Typen instantiiert werden.

Abhilfe: Hüllklassen mit Auto- und Unboxing

Während eine Sammlung mit einem Iterator durchlaufen wird, darf sie nicht verändert werden. (in ihrer Struktur)

Sonst: ConcurrentModificationException

Ausnahme: Änderungen durch den Iterator selbst (wie "remove").