Inhalt

Begriffsbildung: (Endliche) Folge

Unter einer Folge verstehen wir eine „Aufzählung“ von Elementen in einer bestimmten Reihenfolge.
Andere gebräuchliche Bezeichnungen: Sequenz oder Liste

Beachte:

- Elemente können in einer Folge mehrfach vorkommen.
- Die Reihenfolge ist wesentlich!
- Wir sprechen vom „Element an Position x“.
  Die Zählung der Positionen beginnt bei 0. Operationen, die einen Index als Parameter erwarten, können eine IndexOutOfBoundsException auslösen.

Folgen sind ihrer Natur nach „iterierbar“,
sie implementieren das Interface Iterable<T> .

Methoden des ADT Folge (1)

- **int** size ()
  liefert die Anzahl der Elemente
- **boolean** isEmpty()
  prüft, ob die Folge leer ist
- **boolean** contains(T e)
  prüft, ob es ein Element mit Eintrag *e* in der Folge gibt
- T get(**int** i)
  liefert das Element an der *i*-ten Position
- **void** set(**int** i, T e)
  ändert das Element an der *i*-ten Position
- **int** pos(T e)
  liefert die Position (des ersten Vorkommens) von Eintrag *e*
  Löst eine NoSuchElementException aus, falls *e* nicht in der Folge vorkommt.

Methoden des ADT Folge (2)

- **void** insert (**int** i , T e)
  fügt ein neues Element an Position *i* ein; alle folgenden Elemente verschieben sich um eine Position nach hinten
- **void** addFirst (T e)
  fügt eine neues Element vorne in die Folge ein
- **void** addLast(T e)
  hängt eine neues Element an die Folge an
- **void** delete (T e)
  entfernt das (erste Vorkommen von) Element *e* aus der Folge; alle nachfolgenden Elemente verschieben sich um eine Position nach vorne;
  die Methode macht nichts, falls *e* nicht in der Folge vorkommt
- **void** remove(**int** i)
  entfernt das Element an Position *i* aus der Folge; alle nachfolgenden Elemente verschieben sich um eine Position nach vorne

Inhalt

Implementierungen

Wie schon häufig, bieten sich zwei grundsätzlich verschiedene Implementierungsansätze an:

- Array-basiert mittels einer Variante von Dynamischen Arrays
- Referenz-basiert mittels verketterter Listen

Folge als DynArray

- In einer Folge betrachten wir die Elemente als „lückenlos“ durchnummeriert.
- Es ist zB nicht möglich, die „dritte“ Position einer Folge zu belegen, wenn es kein „zweites“ Element gibt.
- Damit ist der ADT Folge die „natürliche“ Abstraktion von Dynamischen Arrays bzw
- damit ist ein dynamisches Array die „natürliche“ Implementierung einer Folge.

Folge als DVL

Alternativ: Implementierung mittels einer „DVL“
Eine DVL (doppelt verkettete Liste) ist ähnlich aufgebaut wie eine EVL, mit folgenden Unterschieden:

- Ein Listenelement besteht nun aus drei Komponenten
  - T value: einer Referenz auf den eigentlichen Datensatz
  - ListenElem prev: einer Referenz auf das vorhergehende Listenelement
  - ListenElem next: einer Referenz auf das nächste Listenelement
- Die DVL erhält ein Referenz auf das erste und auf das letzte Listenelement (first bzw last).

Schematische Darstellung