Übungen zur Vorlesung
Objektorientierte Programmierung und Modellierung

WS 2008/2009

UNIVERSITÄT KOBLENZ · LANDAU
Institut für Informatik

Prof. Dr. Ralf Lämmel
Dr. Manfred Jackel

Übungsblatt 04

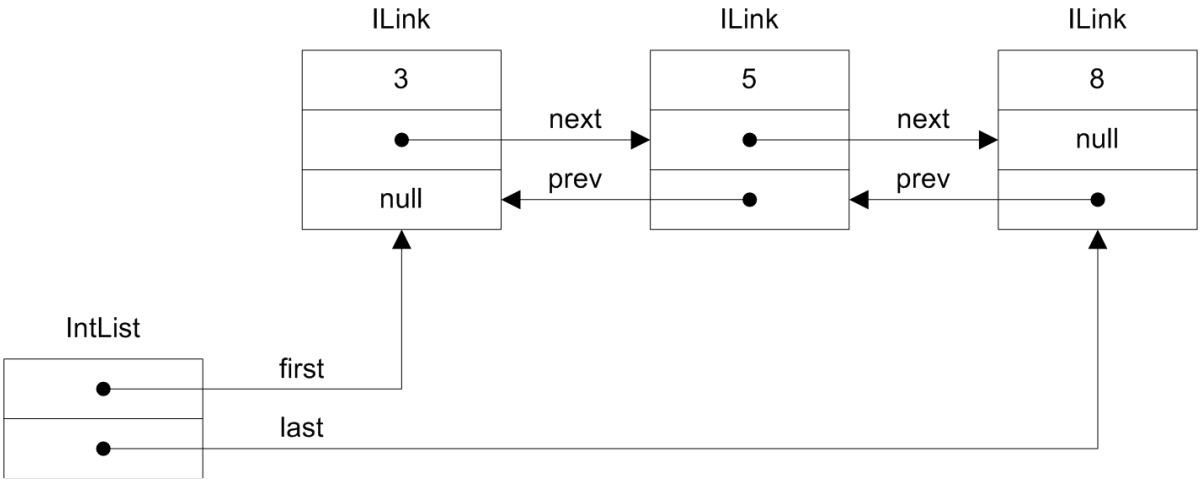
Zu bearbeiten bis 27.11.2008, 24h

Programmieraufgabe 4

Wir skizzieren folgenden Datentyp `IntList`, der eine Liste sortierter Integer-Zahlen implementiert. Die Liste soll doppelt verkettet sein, damit sie leicht in beide Richtungen durchgegangen werden kann. Der Datentyp besteht aus 2 Klassen, `IntList` und `ILink`.

```
public class ILink {
     int value;
     ILink next;
     ILink prev;
}

public class IntList {
     ILink first; // zeigt auf den Listenanfang
     ILink last; // zeigt auf das Listenende

     // Erzeugt eine leere Liste
     public IntList() {
          first = null;
          last = null;
     }

     // Sortiert i an der richtigen Stelle ein
     // (aufsteigende Sortierung)
     public void insert(int i) {
          // Hier Ihre Implementation
     }

     // Liefert true <=> i in Liste enthalten
     public boolean contains(int i) {
     // Hier Ihre Implementation
     }

     // Löscht das 1. Listenelement, das i enthält.
     public void delete(int i) {
     // Hier Ihre Implementation
     }
}
```

Hier ein Beispiel:



Implementieren und Testen Sie diese Klasse, indem Sie ein gegebenes unsortiertes Feld von Integer-Werten in die Liste einfügen, so dass die Elemente aufsteigend angeordnet sind. Implementieren Sie auch Testaufrufe für `contains` und `delete`.

Hausaufgabe 4

Entwerfen sie einen Datentyp `Euro`, der exaktes Rechnen mit Geldbeträgen erlaubt. Dazu werden die Geldbeträge intern als `long` Zahlen dargestellt und zwar in Cent. Methoden dieses Datentyps sollen sein:

```
Euro add(Euro e);
Euro sub(Euro e);
Euro mult(int n);
Euro mult(double d);
```

Testen Sie Ihre Implementation, indem Sie eine Zinseszinstabelle für ein Sparguthaben von 100€ bei jährlicher nachschüssiger Verzinsung von 4 % auf 30 Jahre ausgeben. (Nachschüssige Verzinsung heißt, die jährlichen Zinsen werden am Jahresende zum Guthaben addiert.)

Präsenzaufgabe 4

Benutzen Sie für die Lösung dieser Aufgabe `java.util.LinkedList`.

Natürliches 2–Wegemischen:

Eine Liste wird so sortiert, indem sortierte Teilfolgen (so genannte „Runs“) zu immer längeren sortierten Teilfolgen verschmolzen werden. Dazu werden die sortierten Teilfolgen abwechselnd auf 2 neue Listen verteilt. Beim Zusammenführen dieser beiden Listen werden je 2 Runs der beiden Listen zu einem längeren gemischt.

Hier ein Beispiel:

[0, 8, 9, 7, 15, 13, 11, 1, 19, 14, 17, 17, 13, 2, 15, 4, 4, 15, 1, 0]

Initiales Split
[0, 8, 9, 13, 1, 19, 13, 4, 4, 15, 0]
[7, 15, 11, 14, 17, 17, 2, 15, 1]

Merge & Split
[0, 7, 8, 9, 13, 15, 2, 13, 15, 0]
[1, 11, 14, 17, 17, 19, 1, 4, 4, 15]

Merge & Split
[0, 1, 7, 8, 9, 11, 13, 14, 15, 17, 17, 19, 0]
[1, 2, 4, 4, 13, 15, 15]

Merge & Split
[0, 1, 1, 2, 4, 4, 7, 8, 9, 11, 13, 13, 14, 15, 15, 15, 17, 17, 19]
[0]

Merge & Split
[0, 0, 1, 1, 2, 4, 4, 7, 8, 9, 11, 13, 13, 14, 15, 15, 15, 17, 17, 19]
[]

Abbruch, weil eine der beiden Listen leer ist

Ergebnis: [0, 0, 1, 1, 2, 4, 4, 7, 8, 9, 11, 13, 13, 14, 15, 15, 15, 17, 17, 19]

Abzugeben bis Donnerstag, 27.11.2008, 24h in Gruppenverzeichnis auf
https://svn.uni-koblenz.de/oopm0809/students