Aufgabenblatt 6

Algorithmen und Datenstrukturen, SS 2015

Prof. Dr. Ulrich Hedtstück, HTWG Konstanz

1. Aufgabe

Entwickeln Sie eine rekursive Java-Methode

```
int rmeth(int left, int right);
```

die in einem Array von Integer-Zahlen **mit Hilfe des Teile-und-herrsche-Prinzips** alle negativen Zahlen durch O ersetzt und die Anzahl, wie viele Zahlen negativ waren, als Ergebnis zurückliefert.

Testen und demonstrieren Sie diese Methode mit einem Array der Länge 20 und verwenden Sie dazu Zufallszahlen.

2. Aufgabe

Gegeben sei die Klasse `IntList` für eine verkettete Liste von Integer-Zahlen.

```
public class Node {
  public int value;
  public Node next;
  public Node() {
    value = 0;
    next = null;
  }
  public Node(int v) {
    value = v;
    next = null;
  }
}
```

```
public class IntList {
  public Node head;
  public IntList() {
    head = null;
  }
}
```

Hinweis: Um die Mechanismen der Listenverarbeitung einfacher programmieren zu können, sind hier alle Attribute als `public` *deklariert. Besser wäre, sie als* `private` *zu deklarieren und getter- und setter-Methoden zu verwenden. Wer möchte, kann dies in seiner Lösung berücksichtigen.*

zu Aufgabe 2:

Entwickeln Sie eine Java-Methode

```
public static void listInsert(IntList list1, IntList list2) {...}
```

die die Elemente aus einer gegebenen unsortierten Liste `list1` der Reihe nach in eine zweite, zu Beginn leere Liste `list2` einfügt, und zwar so, dass `list2` immer sortiert bleibt.

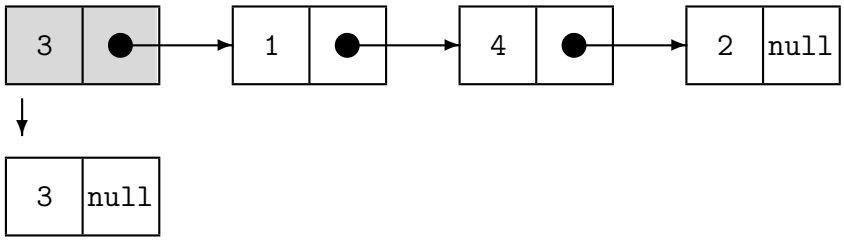
Um das Programm zu testen und auf dem Bildschirm vorzuführen, erzeugen Sie eine Liste `list1` mit 20 Zufallszahlen als int-Werte zwischen 40 und 80, für die anschließend die sortierte Liste `list2` erstellt wird. Geben Sie beide Listen auf dem Bildschirm aus.

Bitte beachten Sie die Anleitung zur Lösung dieser Aufgabe auf der nächsten Seite!

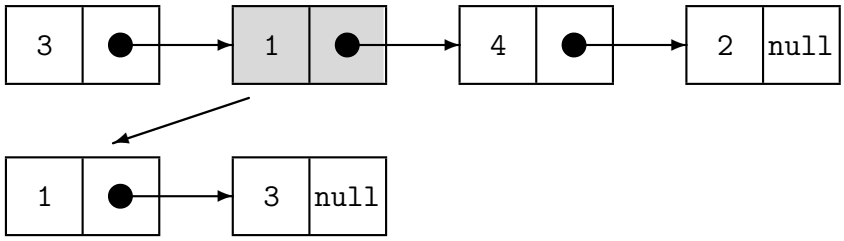
Die Programme sollen in den Übungen vorgeführt und erklärt werden, spätestens am Freitag, den 03.07.2015.

Vorgehensweise Aufgabe 2

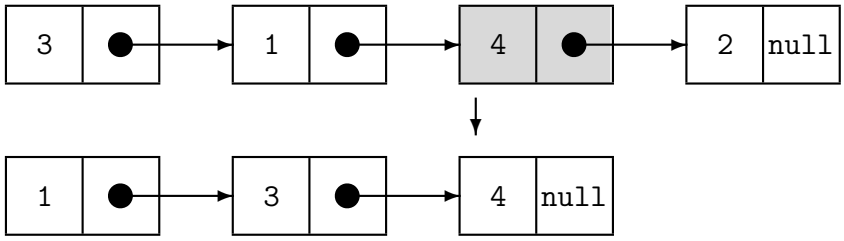
Erzeugung 1. Knoten von `list2`



Erzeugung 2. Knoten von `list2`



Erzeugung 3. Knoten von `list2`



Erzeugung 4. Knoten von `list2`