



Aufgabenblatt 8

Aufgabe 1 (Einfach verkettete Liste, 12 Punkte)

Auf der Homepage der Veranstaltung finden Sie den Quelltext einer generischen Klasse `EVListe` für einfach verkettete Listen. Als Instanzmethoden dieser Klasse stehen zur Verfügung:

- `void append(T elem)`
Hängt ein Element `elem` an das Ende der Liste.
- `T getFirst()`
Liefert das erste Element der Liste als Ergebnis.
- `void remove()`
Löscht das erste Element der Liste.

Hinweis: Die Methoden `getFirst()` und `remove()` prüfen nicht, ob die Liste nicht leer ist. Daher kommt es zu einer `NullPointerException`, wenn eine dieser Methoden auf einer leeren Liste aufgerufen wird.

- Erweitern Sie die Klasse `EVListe` um die Methode
`void insert(T elem)`
die ein Element `elem` am Anfang der Liste (vor dem ersten Element) einfügt.
- Erweitern Sie die Klasse `EVListe` um die Methode
`void remove2()`
die das zweite Element der Liste löscht.
- Erweitern Sie die Klasse `EVListe` um die Methode
`void remove(int k)`
die das k -te Element der Liste löscht.
- Erweitern Sie die Klasse `EVListe` um die Methode
`int size()`
die die Anzahl der Listenelemente als Ergebnis liefert.
- Erweitern Sie die Klasse `EVListe` um die Methode
`int sizeR()`
die ebenfalls die Anzahl der Listenelemente als Ergebnis liefert, für die Berechnung aber keine Schleife verwendet.
Hinweis: Nutzen Sie Rekursion.
- Schreiben Sie ein Testprogramm, um Ihre Methoden aus (a) bis (e) zu testen.

Aufgabe 2 (Geschlossene doppelt verkettete Liste, 10 Punkte)

Das *Josephus-Problem* (benannt nach dem jüdischen Historiker Flavius Josephus) ist wie folgt definiert: Es werden n nummerierte Objekte im Kreis angeordnet. Dann wird, beginnend mit der Nummer k , jedes k -te Objekt entfernt, wobei der Kreis immer wieder geschlossen wird. Ziel ist es, zu gegebenem n und k das letzte Objekt zu bestimmen, das übrig bleibt.

Beispiel: Es sei $n = 5$ und $k = 2$. Die fünf Objekte bilden einen Kreis:

```

    1
  5   2
  4   3
  
```

Zuerst wird das Objekt 2 gelöscht:

```

    1
  5   3
    4
  
```

Dann das Objekt 4:

```

    1
  5   3
  
```

Geht man wieder $k = 2$ Objekte weiter, kommt man zur 1, die nun gelöscht wird:

```

  5   3
  
```

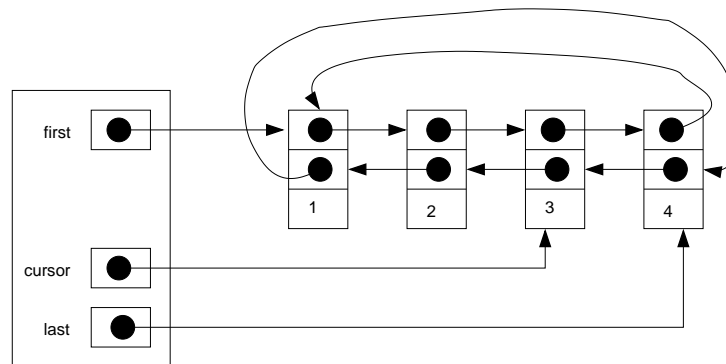
Zwei Objekte weiter liegt nun die 5, die gelöscht wird:

```

  3
  
```

Damit bleibt die 3 übrig und ist die Lösung des Josephus-Problems für $n = 5$ und $k = 2$.

Schreiben Sie ein Java-Programm zur Lösung des Josephus-Problems für beliebiges n und k . Stellen Sie dabei den Kreis der nummerierten Objekte als doppelt verkettete *geschlossene* Liste dar. In einer geschlossenen Liste verweist der `next`-Zeiger des letzten Elements wieder auf das erste Element und der `prev`-Zeiger des ersten Elements auf das letzte Element.



Überlegen Sie sich, wie die Operationen `insert()` und `delete()` für geschlossene Listen implementiert werden müssen.

Hinweis: Für die Lösung des Josephus-Problems sind die Instanzvariablen `first` und `last` nicht notwendig, die Instanzvariable `cursor` ist ausreichend. Es bietet sich aber an, eine Instanzvariable `size` und eine Methode

```
public int size()
```

vorzusehen, mit der die aktuelle Größe einer Liste ermittelt werden kann.

Hilfe zur Lösung der Aufgaben erhalten Sie in den Übungen der KW 23. Abgabe der Lösungen bis **Sonntag, 9. Juni 2013, 24 Uhr** via E-Mail an die Adresse `dat-alg13@mail.inf.h-brs.de`.