technische universität
dortmund

Fakultät für Informatik
Lehrstuhl 14
Lars Hildebrand, Alexander Lochmann,
Iman Kamehkhosh, Marcel Preuß, Dominic Siedhoff

Praktikum zu

Einführung in die Informatik für LogWiIngs und WiMas

Wintersemester 2016/17

Übungsblatt 11

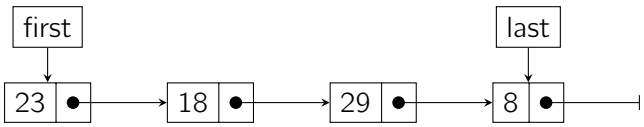
Besprechung:
30.01–
03.02.2017
(KW 5)

Vorbereitende Aufgaben

Aufgabe 11.1: Listen Zeichnen

Dieser Übungszettel beschäftigt sich mit der Datenstruktur der Liste.

Folgendes Diagramm zeigt eine verkettete Liste mit Kopf (first) und Fußzeiger (last):



Führen Sie auf dieser Liste folgende drei Operationen hintereinander aus und repräsentieren Sie dabei die nach jedem Schritt entstandene Liste wie oben:

- Entfernen des Elements mit der Zahl 29
- Anfügen der Zahl 12 an das Ende der Liste
- Entfernen des Elements mit der Zahl 23

Präsenzaufgaben

Aufgabe 11.2: Listen – Implementierung 1

In dieser Aufgabe sollen Sie eine Liste implementieren, die **int**-Werte verwalten kann.

Das erste Element hat dabei, wie bei Arrays, den Index 0, das zweite den Index 1, usw.

a)
- Erstellen Sie die neuen Klassen **List** und **Element** im Paket **blatt11**.
- Die Klasse **List** besitzt zwei Referenzen auf Elementobjekte mit den Namen **first** und **last** für Kopf und Fuß. Im Konstruktor sollen beide auf **null** gesetzt werden. Der Konstruktor besitzt keine Parameter.
- Die Klasse Element soll zwei Attribute besitzen: Einen Zahlenwert **value** und eine Referenz **next** auf das nächste Element.

  Der Konstruktor eines Elements muss entsprechend einen Wert entgegennehmen und sein **value**-Attribut setzen. Die Referenz auf das nächste Element ist bei Instanziierung zunächst immer **null**.

b) Implementieren Sie anschließend folgende Methoden:
- Die Klasse **Element** erhält eine Methode **setNext**, die das nächste Element auf ein übergebenes Element setzen soll, und eine Methode **getNext**, die dieses zurückgibt.
- Die Klasse **List** erhält eine Methode **add**, die einen **int** entgegennimmt um ein neues Element zu erzeugen und an das Ende der Liste anzufügen. Beachten Sie dabei die Referenzen des Kopfes und des Fußes entsprechend anzupassen. Ebenso muss die **next**-Referenz des aktuell letzten Elementes angepasst werden, falls es existiert.

c) Es fehlt die Möglichkeit, Daten aus der Liste auszulesen:

Implementieren Sie eine Methode **get**, die einen **int** *n* entgegennehmen soll und den **Wert** des Elementes mit dem Index *n* aus der Liste zurückgibt.

Gehen Sie davon aus, dass nur auf vorhandene Indizes zugegriffen wird.

Testen Sie Ihre Implementierung z. B. an dieser Testklasse:

```
1  package blatt11;
2
3  public class ListTest1 {
4      public static void main(String[] args) {
5          List list = new List();
6
7          list.add(47);
8          list.add(237);
9          list.add(9);
10
11         if(list.get(0) != 47
12         || list.get(1) != 237
13         || list.get(2) != 9) {
14             System.out.println("There are definitely errors in your code");
15         } else {
16             System.out.println("All values seem to be in their place");
17         }
18     }
19 }
```

Aufgabe 11.3: Listen – Implementierung 2

a) Fügen Sie eine Möglichkeit hinzu, Daten aus der Liste zu entfernen:

Implementieren Sie eine Methode **remove**, die einen **int** *n* entgegennehmen soll und das Element mit dem Index *n* aus der Liste entfernt.

Dabei soll der Wert des entfernten Objektes zurückgegeben werden.

Beachten Sie dabei, die passenden Referenzen zwischenzuspeichern, um die Verkettung der Liste nicht zu unterbrechen.

b) Es sollte noch möglich sein, die Größe einer Liste zu erfahren:

Machen Sie sich Notizen darüber, wie Sie die Größe einer Liste errechnen können. Machen Sie sich dann Notizen darüber, wie sie die Größe einer Liste während ihrer Verwendung abspeichern und manipulieren können. Gibt es Vorteile bei einer Lösung? Gibt es signifikante Nachteile bei einer Lösung?

---

c) Testen Sie Ihre Implementierung z. B. an dieser Testklasse:

```
1  package blatt11;
2
3  public class ListTest2 {
4      public static void main(String[] args) {
5          List list = new List();
6
7          list.add(47);
8          list.add(237);
9          list.add(9);
10         list.add(9001);
11
12         if(list.getSize() != 4) {
13             System.out.println("You have to work on your size function");
14         }
15
16         if(list.remove(0) != 47
17         || list.remove(1) != 9
18         || list.remove(1) != 9001
19         || list.get(0) != 237) {
20             System.out.println("There are definitely errors in your code");
21         } else {
22             System.out.println("Removing seems to work");
23         }
24
25         if(list.getSize() != 1) {
26             System.out.println("You have to work on your size function");
27         }
28     }
29 }
```

Ergänzende Aufgaben

Aufgabe 11.4: Liste von Objekten

Schreiben Sie eine alternative Implementierung der Liste, die in der Lage ist, Objekte wie z. B. **Vehicle** aus Blatt 10 zu verwalten.