

Praktikum zu
**Einführung in die Informatik für
LogWings und WiMas**
Wintersemester 2014/15

Übungsblatt 12

Bearbeitungszeit:

26.01.15 -

30.01.15

Aufgabe 12.1 – Listen

Bearbeitungszeit: 15 Minuten

- Wenden Sie die nachfolgend aufgelisteten Operationen auf eine (anfängs leere) Liste an. Stellen Sie die Liste dabei vor und nach jeder Operation grafisch dar und markieren Sie dabei jeweils Kopf und Fuß der Liste. Achten Sie zudem auf die Reihenfolge, in der die Zeiger modifiziert werden!
 - Am Ende einfügen: 4
 - Am Ende einfügen: 7
 - Zwischen dem ersten und zweiten Element einfügen: 2
 - Am Anfang einfügen: 6
 - Das dritte Element entfernen
 - Element mit Wert 7 entfernen
- Skizzieren Sie anschließend eine Liste, welche Referenzen auf vier verschiedene Objekte vom Typ **Contact** aus Übungsblatt 11 verwaltet. Verwenden Sie als Skizze für ein Kontaktobjekt beispielsweise folgende Repräsentation:

Max Mustermann Am Beispiel 42

Aufgabe 12.2 – Listenprogrammierung

Bearbeitungszeit: 60 Minuten

Gegeben sei folgende Klasse zur Verwaltung von Listenelementen:

```
1 package blatt12;
2
3 public class Element {
4     private int value;
5     private Element next;
6     public Element ( int value ) {
7         this.value = value;
8         this.next = null;
9     }
10    public int getValue(){
11        return this.value;
12    }
13    public Element getNext(){
14        return this.next;
15    }
16    public void setNext(Element next){
17        this.next = next;
18    }
19 }
```

- Ändern Sie die Klasse so um, dass sie nun Werte vom Typ **Contact** verwaltet.
 - Erstellen Sie eine Klasse **ContactList**, welche folgende Attribute besitzt:
 - Ein privates Attribut **head** vom Typ **Element**.
 - Ein privates Attribut **last** vom Typ **Element**.
 - Ein privates Attribut **size** vom Typ **int**.
 - Der Konstruktor der Klasse soll keinen Parameter entgegennehmen und alle Attribute auf **null**, bzw. 0 setzen.
 - Implementieren Sie eine Methode **public boolean isEmpty()**, welche überprüfen soll, ob die Liste leer ist.
 - Implementieren Sie eine Methode **public void add(Contact toAdd)**, welche an das Ende der Liste ein neues Element mit dem Kontakt **toAdd** hinzufügen soll.
 - Implementieren Sie eine Methode **public Contact get(int n)**, welche das n -te Element der Liste zurück geben soll, oder **null**, wenn kein Element an dieser Position existiert.
 - Implementieren Sie hierfür eine private Methode **private Contact get(int n, Element current)**, mit welcher Sie rekursiv durch die Liste laufen sollen.
 - Wie können Sie die Anzahl der in der Liste enthaltenen Elemente herausfinden?
-
- Schreiben Sie eine Methode **public int getSize()**, welche die Größe der Liste zurück gibt.
 - Schreiben Sie eine Methode **public boolean isLast(Contact contact)**, welche überprüfen soll, ob der übergebene Kontakt im letztem Element der Liste steht.
 - Schreiben Sie eine Testklasse, welche Daten in die Liste einfügt und ausgibt. Beispielsweise kann eine Ausgabe der Testklasse folgendermaßen aussehen:

- 0) Max Mustermann
- 1) Anna Anderson
- 2) Linus Torvalds
- 3) John Carmack