

Implementierung der Liste mit den Klassen Liste und Knoten

Mit Methoden zum löschen (`remove(int x)`), hinzufügen (`add()`) und suchen (`getKnoten(int x)`) von Knoten.

In der Klasse Liste:

```
public class Liste
{
    private Knoten anfang;
    private Knoten ende;
    public Liste()
    {
        anfang=null;
        ende=null;
    }
    //Methode zum ermitteln der Größe der Liste, wenn
    es einen Anfang gibt, wird rekursiv mit einer
    Methode der Klasse Knoten gesucht
    public int size(){
        if (anfang==null) {
            return 0;
        } else {
            return anfang.size();
        }
    }
    //Methode zum hinzufügen von Knoten
    public void add(String name)
    {
        //Es wird ein neuer Knoten erzeugt, und dieser wird
        als Anfang deklariert wenn noch kein Anfang
        vorhanden ist, sonst wird der neue Knoten als
        nachfolger eingesetzt
        Knoten k=new Knoten(name);
        if (anfang==null){anfang=k;
        ende=k;}
        else
        {ende.setNachfolger(k);
        ende=k;}
    }
    public void remove()
    {
```

```
        if(anfang==null);
        else
        {
            anfang=anfang.getNachfolger();
        }
    }
    //Methode zum entfernen von Knoten
    public void remove(int i)
    {
        if(anfang==null){
            System.out.println("Fehler");
        } else if ((size()==1) &&
        (i==0)){ remove();}
        else {
            //Es wird der vordere des zu löschende Knotens
            ermittelt, und der hintere des zu löschenden wird
            diesem als Nachfolger gegeben, somit ist der
            gelöschte Platz „überbrückt“
            Knoten vK = getKnoten(i-1);
            Knoten hK = getKnoten(i+1);
            vK.setNachfolger(hK);
        }
    }
    public void listeLeeren()
    {
        anfang=null;
    }
    //Methode zum Suchen von Knoten
    public Knoten getKnoten(int x)
    {
        if (anfang==null) return
        null;
        else if (x<0) return null;
```

```
        else if (x>=size()) return
        null;
        else return
        anfang.getKnoten(x);
    }
}
```

In der Klasse Knoten:

```
public class Knoten
{
    private String name;
    private Knoten nachfolger;
    //Hier sind die zugehörigen Methoden von „Knoten“
    für die Klasse „Liste“ enthalten
    public Knoten(String n){ name=n;
    nachfolger=null;}
    public void setNachfolger(Knoten
    k){nachfolger=k;}
    public void anzeigen()
    {System.out.println(name);}
    public Knoten getNachfolger()
    {return nachfolger;}
    public int size(){
        if (nachfolger==null){ return 1;}
        else
        { return
        nachfolger.size()+1;}
    }
    public Knoten getKnoten(int x){
        if (x==0) return this;
        else return
        nachfolger.getKnoten(x-1);
    }
}
```