

## Aufgabe 1

- a) Zweites Element löschen:  
toFirst – next – remove
- b) Zwischen erstem und zweitem Element einfügen:  
toFirst – next – insert
- c) hasAccess() false:  
- wenn die Liste leer ist  
- wenn die Liste nicht leer ist, aber toFirst noch nicht aufgerufen wurde  
- wenn toFirst und so oft next aufgerufen wurde, dass current nicht mehr auf ein Element zeigt  
- letztes Element ausgewählt, dann remove  
hasAccess() true?  
- wenn die Liste nicht leer ist, toFirst und ggf. mehrmals next aufgerufen und current noch auf ein Element zeigt
- d) Insert() fügt VOR dem aktuellen Element ein, und NICHT, wenn kein Element ausgewählt ist.  
Daher kann man mit insert() maximal als vorletztes Element einfügen.  
Um am Ende der Liste einzufügen kann man (jederzeit) append() benutzen.

## Aufgabe 3

```
/*
 * Vertauscht Elemente einer Liste
 * (Name, Datum)
 */
public class Aufgabe3
{
    private List<String> liste;

    public Aufgabe3()
    {
        liste = new List();
    }

    public void fuellen()
    {
        liste.append("Rot");
        liste.append("Blau");
        liste.append("Gelb");
        liste.append("Grün");
        liste.append("Lila");
        liste.append("Braun");
        liste.append("Pink");
        liste.append("Orange");
        liste.append("Grau");
    }

    public void vertauschen()
    {
        String s2, s4;

        liste.toFirst();
        liste.next();
        s2 = liste.getContent();
        liste.next();
        liste.next();
        s4 = liste.getContent();
        liste.insert(s2);
        liste.remove();
        liste.toFirst();
        liste.next();
        liste.insert(s4);
        liste.remove();
    }
}
```

```
public void vertauschen2()
{
    String s2, s4;

    if (!liste.isEmpty())
    {
        liste.toFirst();
        liste.next();
        s2 = liste.getContent();
        liste.next();
        liste.next();
        s4 = liste.getContent();

        if (liste.hasAccess())
        {
            liste.insert(s2);
            liste.remove();
            liste.toFirst();
            liste.next();
            liste.insert(s4);
            liste.remove();
        }
    }
}
```

```
public void vertauschen3(
    int pos1, int pos2)
{
    int i, t;
    String s1, s2;

    // Ungültige Fälle
    if (pos1 < 0 || pos2 < 0 ||
        pos1 == pos2)
    {
        return;
    }
    // pos2 soll nach pos1 kommen
    if (pos1 > pos2)
    {
        t = pos1;
        pos1 = pos2;
        pos2 = t;
    }

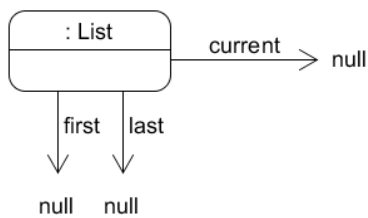
    // Elemente suchen
    liste.toFirst();
    for (i = 0; i < pos1; i++)
    {
        liste.next();
    }
    s1 = liste.getContent();
    for (i = 0; i < pos2 - pos1; i++)
    {
        liste.next();
    }
    s2 = liste.getContent();

    // Nicht genügend Elemente
    if (!liste.hasAccess())
    {
        return;
    }

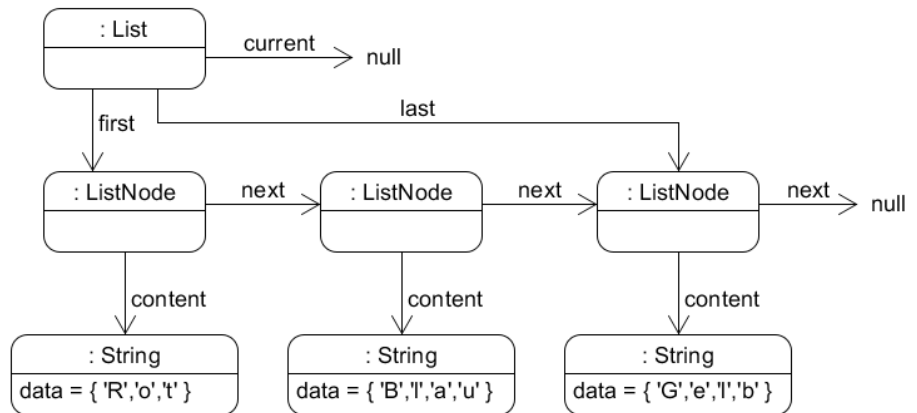
    // Tauschen
    liste.insert(s1);
    liste.remove();
    liste.toFirst();
    for (i = 0; i < pos1; i++)
    {
        liste.next();
    }
    liste.insert(s2);
    liste.remove();
}
```

## Aufgabe 2

Objektdiagramm 1



Objektdiagramm 2



Objektdiagramm 3

Nach toFirst, next, remove, next zeigt current auf null.  
In diesem Fall funktioniert insert nicht (es geschieht nichts)  
"Grün" wird also nicht eingefügt.

