

## Zu 36.1 & 36.2 – Generische Typen

Die Beispiele und Selbsttestaufgaben aus dem Übungsheft zu `LinkedList()` und `iterator()` zusammen mit Kommentaren. Methoden mit verschiedenen Versionen sind als Kommentarblöcke gekennzeichnet.

```
import java.util.*;
import java.util.LinkedList;

public class GenerischeKlasse {

    /* ____print____ mit while-Schleife
    *
    public void print(Collection<String> meineCollection) { // Collection<String>

        //Abbruchbedingung / Standardfall
        if(meineCollection == null) {
            return;
        }

        //Iterator erzeugen
        Iterator<String> meinItterator = meineCollection.iterator();

        //Alle Elemente von meineCollection mittels meinItterator durchlaufen
        while(meinItterator.hasNext()) {
            String sTemp = meinItterator.next();

            System.out.println("print: " + sTemp);

        }

        //Nach jeder Ausgabe wird das letzte Element der Liste entfernt
        meinItterator.remove();
    }
    */

    // print mit for-Schleife
    /*
    public void print(Iterable<String> paraItt) {
        if(paraItt == null) {
            return;
        }

        for(Iterator<String> myItt = paraItt.iterator(); myItt.hasNext();) {
            String sTemp = myItt.next();

            System.out.println("print: " + sTemp);

        }

    }
    */
}
```

```

//print mit erweiterter for-Schleife
public void print (Iterable<String> myIt) {
    if(myIt == null) {
        return;
    }

    for(String s : myIt) {
        System.out.println(s);
    }
}

public void längeZeichenketten(List<String> myList) {

    if(myList == null) {
        return;
    }

    //Startwerte = Länge 1. Element
    Iterator<String> myIterator = myList.iterator();
    int max = myIterator.next().length();
    int min = max;
    System.out.println("maxInit: " + max);
    System.out.println("minInit: " + min);

    for(String sTemp : myList) {
        if(sTemp.length() > max) {
            max = sTemp.length();
        }
        if (sTemp.length() < min) {
            min = sTemp.length();
        }
    }
    System.out.println("Maximum: " + max);
    System.out.println("Minimum: " + min);
}

```

```

public void findeZeichenketten(List<String> myList) {

    if(myList == null) {
        return;
    }

    String max = myList.get(0);
    String min = myList.get(0);

    for(String sTemp : myList) {
        if(sTemp.length() > max.length()) {

```

```

        max = sTemp;
    }
    if(sTemp.length() < min.length()) {
        min = sTemp;
    }
}

System.out.println("Min: " + min);
System.out.println("Max: " + max);
}

```

```

public static void main(String[] args) {

    GenerischeKlasse genero = new GenerischeKlasse();

    /*Liste mit beliebigen Objekten String/Integer gemischt
    //neue Liste erzeugen
    List l = new LinkedList();
    //Zeichenkette hinzufügen
    l.add("Hallo");
    //Zahl hinzufügen
    l.add(3);
    //Integer Objekt wird autom. erzeugt

    Object o = l.get(0);

    System.out.println(l);
    */

    //neue Liste erzeugen
    List<String> ls = new LinkedList<String>();
    List<Integer> li = new LinkedList<Integer>();
    //Zeichenkette hinzufügen
    ls.add("Hallo1");
    ls.add("Hallo22");
    ls.add("Hallo333");
    ls.add("Hallo4444");
    ls.add("Hallo");
    //Zahl hinzufügen
    li.add(3);
    //Integer Objekt wird autom. erzeugt

    String zeichenkette = ls.get(0);
    int zahl = li.get(0);
    //Werte der String & Integer Objekte werden Variablen vom Typ String/int
    zugewiesen --> quasi umgewandelt

    li.add(4);
    li.add(3);
    li.add(5);

    List<Integer> teilliste = li.subList(1, 3);

```

```
/*System.out.println("li " + li);
System.out.println("zeichenkette " + zeichenkette);
System.out.println("zahl " + zahl);
System.out.println("teilliste Stellen 2-3" + teilliste);
```

```
System.out.println("ls4 " + ls);
genero.print(ls);
System.out.println("ls3 " + ls);
genero.print(ls);
System.out.println("ls2 " + ls);
genero.print(ls);
*/
```

```
System.out.println("ls1 " + ls);
genero.print(ls);
genero.längeZeichenketten(ls);
genero.findeZeichenketten(ls);
```

```
}
```

```
}
```