



# Collections

Lösungen zu den  
Übungsaufgaben

```
/**
 * Erzeugen einer Linked-List aus Zahlen, Ausgabe auf der Konsole.
 */
public class ListeUebung {

    /**
     * Programmeinstieg.
     */
    public static void main(String[] args) {
        List<Integer> liste =
            new LinkedList<Integer>(Arrays.asList(new Integer(
                23), new Integer(42), new Integer(12),
                new Integer(11)));

        String ergebnis = "{";
        for (int i = 0; i < liste.size(); i++) {
            ergebnis +=
                String.format("%d"
                    + ((i < liste.size() - 1) ? ", " : ""),
                    liste.get(i));
        }
        ergebnis += "}";
        System.out.println(ergebnis);
    }
}
```

```
// Durchlaufen mit for-Schleife
for (Iterator<Integer> it = liste.iterator(); it
    .hasNext();) {
    System.out.println(it.next());
}
```

```
// Durchlaufen mit while-Schleife
Iterator<Integer> it = liste.iterator();
while (it.hasNext()) {
    System.out.println(it.next());
}
```

```
class Bruch implements Comparable<Bruch> {  
  
    ...  
  
    @Override  
    public int compareTo(Bruch anderereBruch) {  
        double differenz = getWert() - anderereBruch.getWert();  
        if (differenz < -1e-5) {  
            return -1;  
        } else if (differenz > 1e-5) {  
            return 1;  
        } else {  
            return 0;  
        }  
    }  
}
```

# Übung: Menge und Map



```
/**
 * In der Bank werden Kontostände durch (eindeutige) Nummern referenziert.
 */
public class SchweizerNummernKontoBank {
    /**
     * Liste der Konten als Schlüssel-Wert-Paare. Die Kontonummer ist der
     * Schlüssel
     */
    private Map<Integer, Float> konten =
        new HashMap<Integer, Float>();

    /**
     * Neues Konto hinzufügen. Achtung: keine Überprüfung, ob die Kontonummer
     * bereits verwendet wird.
     */
    public void kontoHinzufueegen(int kontonummer,
        float kontostand) {
        konten.put(new Integer(kontonummer), new Float(
            kontostand));
    }

    /**
     * Liefert den aktuellen Kontostand für eine gegebene Kontonummer;
     */
    public float getKontostand(int kontonummer) {
        return konten.get(kontonummer);
    }

    /**
     * Neuen Kontostand für ein Konto setzen. Achtung: Konto muss bereits
     * existieren.
     */
    public void setzeKontostand(int kontonummer,
        float neuerKontostand) {
        if (konten.get(kontonummer) == null) {
            System.out.println("Ungültige Kontonummer!");
        }
        konten.put(kontonummer, neuerKontostand);
    }
}
```