

A decorative header featuring a complex, multi-colored circuit board pattern in shades of blue, green, and yellow, set against a dark teal background.

# Iterator

Iterator bei Sammlungsklassen

# Iterator

Achtung:

Die Behandlung von Iteratoren ist **überholt**, da inzwischen bei Sammlungsklassen mit

```
for (<Klasse> <objekt> : <Sammlung>
    { ... }
```

über alle Elemente iteriert werden kann.

# Iterator

Wiederholungen (Schleifen) treten nicht nur bei Sammlungsklassen auf.

Bei Sammlungsklassen treten Wiederholungen aber sicher auf, da wir in der Regel nicht nur gezielt auf einzelne Elemente zugreifen wollen, sondern auf alle Elemente.

# Iterator

Sammlungsklassen stellen für Wiederholungen daher eine Methode bereit,

die zu einem Sammlungsobjekt jeweils ein Iteratorobjekt liefert:

Es ist die Methode `iterator()`.

# Iterator

Um ein Iteratorobjekt nutzen zu können, müssen wir das zugehörige Paket importieren:

```
import java.util.Iterator;
```

# Iterator

Ohne eine Typisierung müssen wir  
in der Schleife einen cast  
ausführen:

```
for (Iterator it= schraenke.iterator();  
     it.hasNext();)  
{schrangkand.append(  
  ((Schrang) it.next())  
  .gibAktuelleFigur(),  
  false);
```

# Iterator

Mit Typisierung wird es etwas einfacher:

```
for (Iterator<Moebe1>
    it= schraenke.iterator();
    it.hasNext();)
{schrankwand.append(
    it.next().gibAktuelleFigur(),
    false);
```

# Iterator

„Schleifenvariable“ ist der von der Klasse `ArrayList` bereitgestellte Iterator, den wir durch die Methode `iterator()` bekommen. Laufbedingung ist der Wert der Methode `hasNext()`

Der Iterator steuert den Iterationsschritt selbst; wie er das macht, bleibt ihm selbst überlassen

Wir greifen mit der Methode `next()` auf die Elemente des Iterators zu

# Iterator

Bemerkenswert ist:

Für die Schleife ist es egal, welchen Sammlungstyp wir verwenden.

Es geht also bei

ArrayList

LinkedList

Vector

HashSet

TreeSet

prinzipiell gleich.

Allerdings kann sich die Reihenfolge der iterierten Elemente unterscheiden.