TECHNISCHE UNIVERSITÄT DRESDEN

OOSE5

Java Collection Framework am Beispiel von ToDo-Listen

Lehrstuhl Softwaretechnologie, Dr. Birgit Demuth
Sommersemester 2016

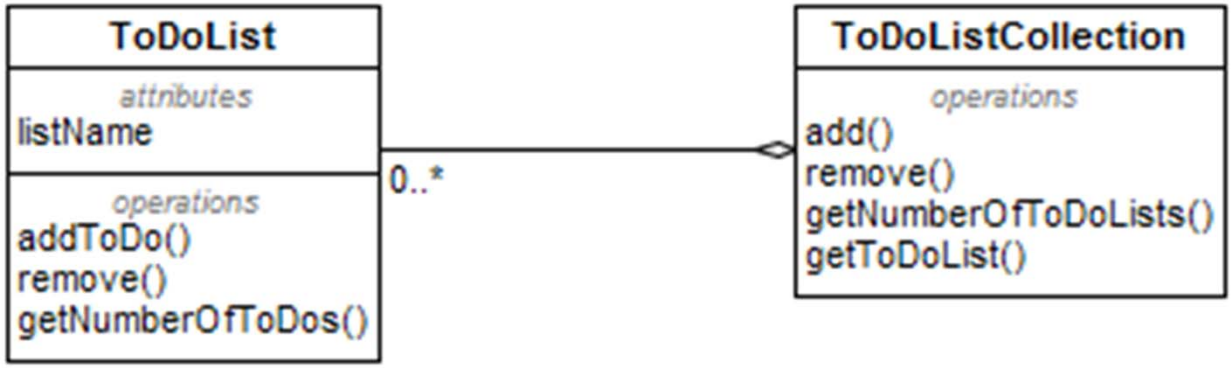
Phasen in der Softwareentwicklung

Phase	Modellierung in UML (Klassendiagramme)
Analyse (OOA)	aUML (UML-Analyseklassendiagramm)
Entwurf (OOD)	dUML (UML-Entwurfsklassendiagramm)
Implementierung und Test (OOP)	jUML (UML-Implementationsklassendiagramm für Java)

Literatur:

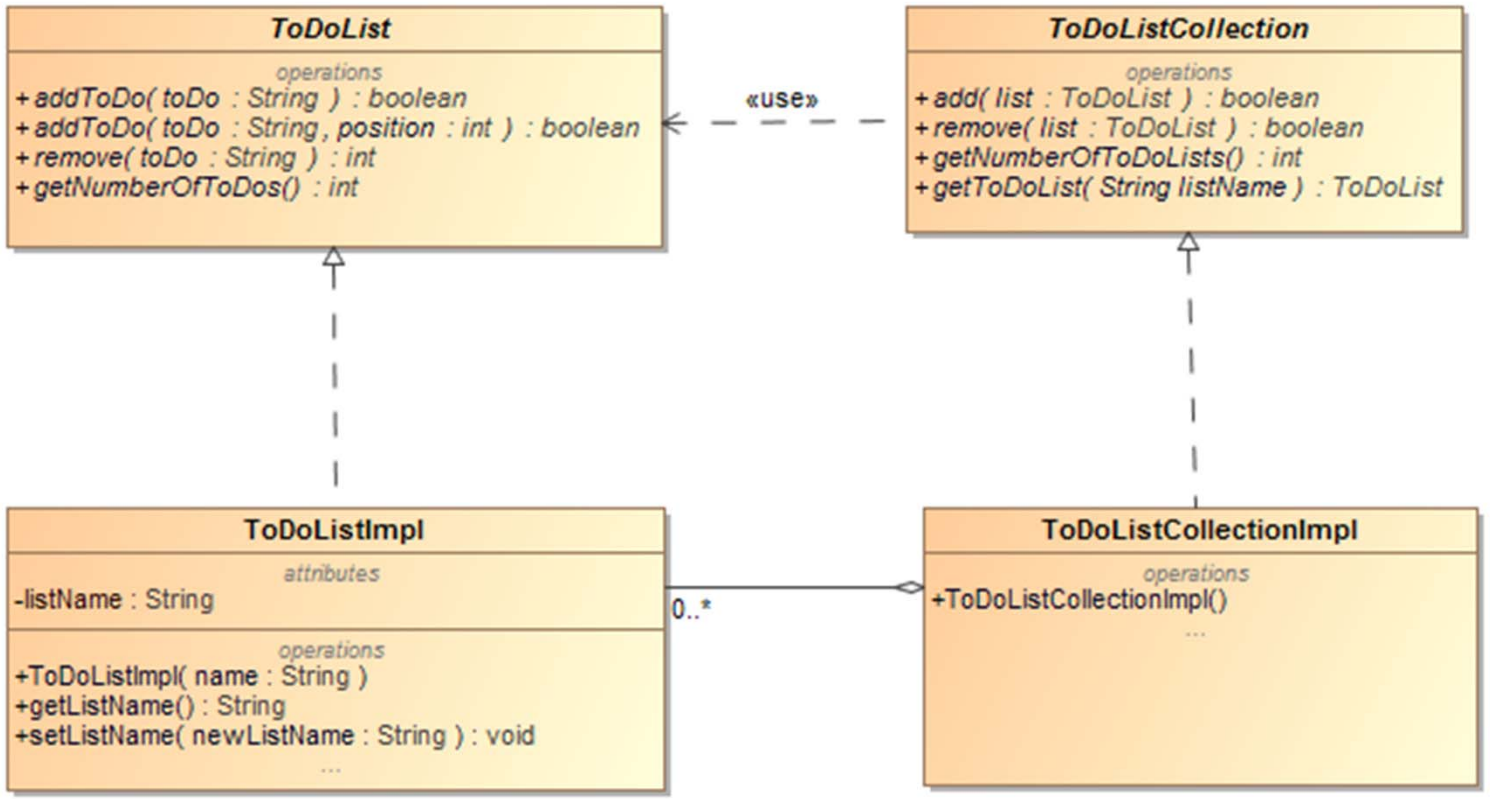
Birgit Demuth (Hrsg.): Softwaretechnologie für Einsteiger.
Pearson Studium, 2. geänderte Auflage, 2014
Abschnitt Klassendiagramme, S. 73 – 112

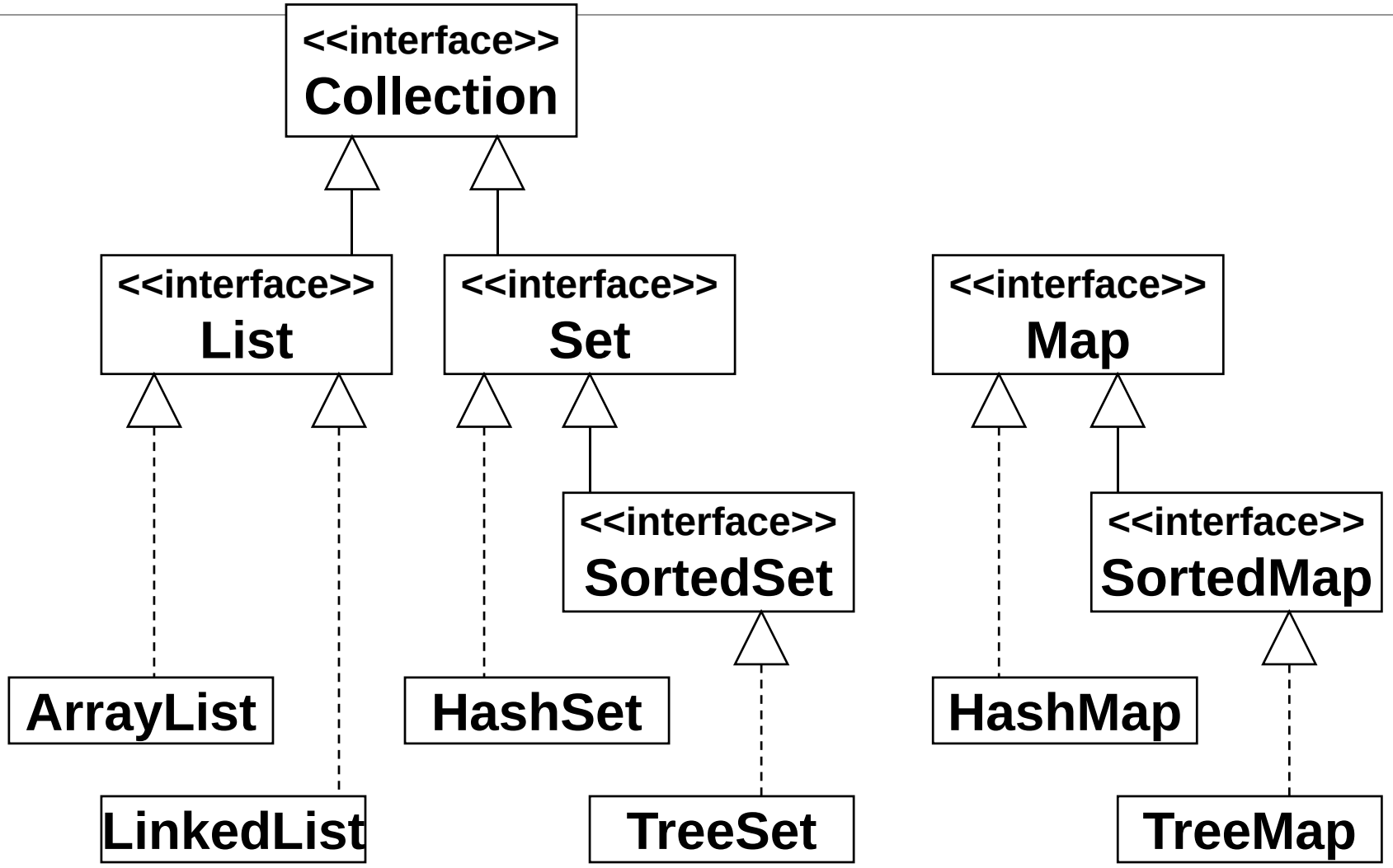
ToDo-Listen (aUML)



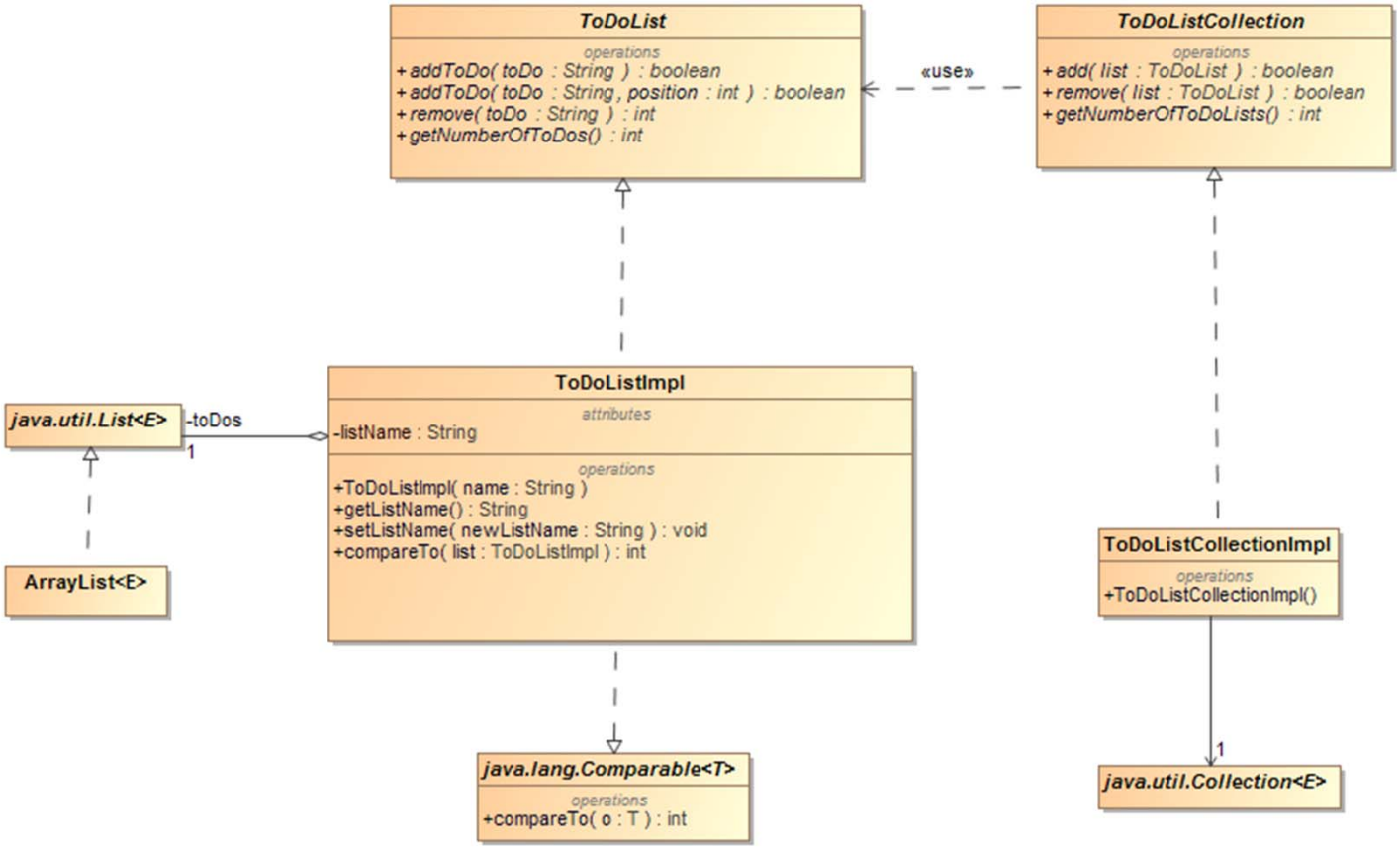
- ToDoList: Eine ToDo-Liste beinhaltet in geordneter Reihenfolge „toDos“.
- ToDoListCollection: Die ToDo-Listen einer Person werden in einer Kollektion verwaltet.
- Über die Methode getToDoList() soll über den Listennamen (listname) auf eine konkrete ToDoListe zugegriffen werden.

ToDo-Listen (dUML)




<<interface>>
Collection
<<interface>>
List
<<interface>>
Set
<<interface>>
Map
<<interface>>
SortedSet
<<interface>>
SortedMap
ArrayList
HashSet
HashMap
LinkedList
TreeSet
TreeMap

ToDo-Listen (jUML) (Klausur WS 2014/15)



java.util.List<E> (Auszug)

https://docs.oracle.com/javase/8/docs/api/java/util/List.html

```
public interface List<E> {

  public boolean add (E o)
  public boolean add (int index, E o)
  public boolean remove (Object o)
  public E remove (int index)
  public void clear()
  public boolean isEmpty()
  public boolean contains (Object o)
  public int size()
  ...
  public boolean equals (Object o)
  public E get(int index)
  ...
  public Iterator<E> iterator()
}
```

java.util.Set<E> (Auszug)

https://docs.oracle.com/javase/8/docs/api/java/util/Set.html

```
public interface Set<E> {

  public boolean add (E o);
  public boolean addAll(Collection<? extends E> c)
  public boolean remove (Object o);
  public void clear();
  public boolean isEmpty();
  public boolean contains (Object o);
  public int size();
  public boolean equals (Object o);
  public int hashCode();
  ...
  public Iterator<E> iterator();
}
```

java.lang.Comparable<T>

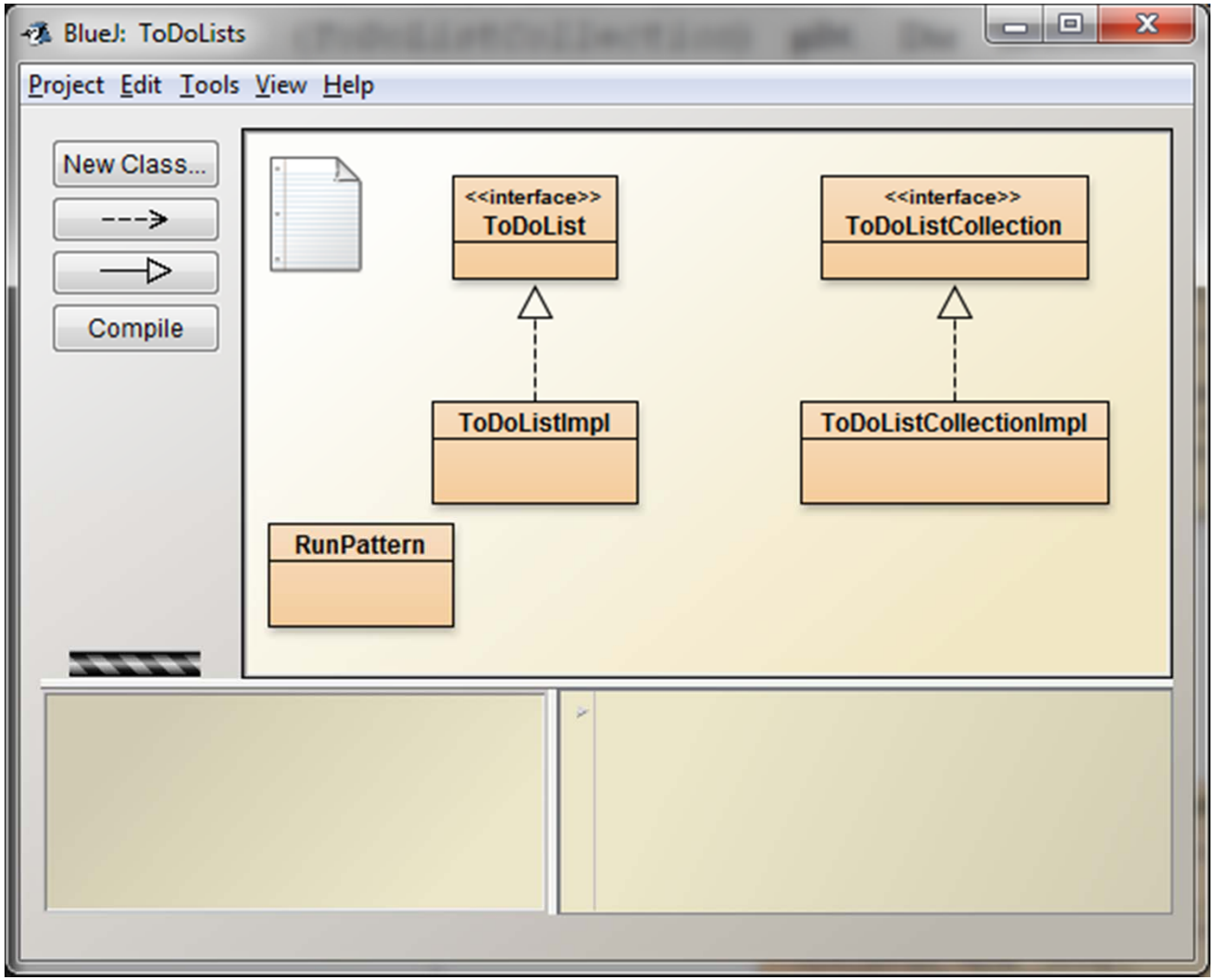
https://docs.oracle.com/javase/8/docs/api/java/lang/Comparable.html

```
public interface Comparable<T> {

   public int compareTo (T o);
}
```

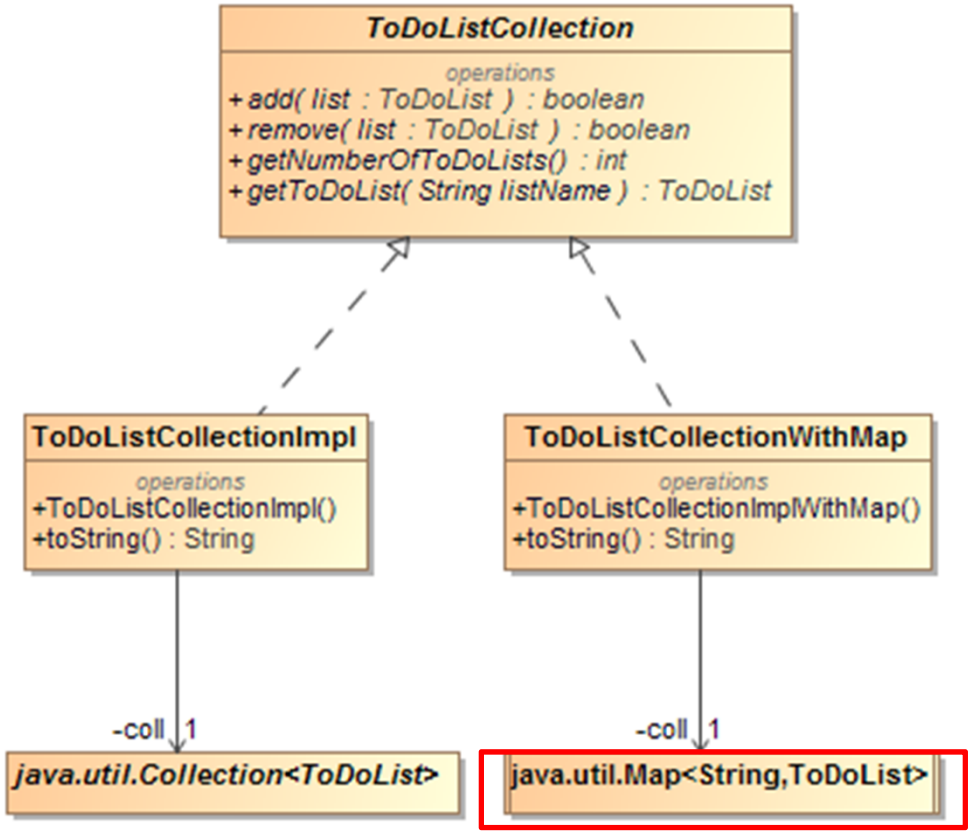
- Interface für den Test der Ordnung auf Elementen
- T ist der Typ des Objektes, mit dem verglichen wird
- Resultat kleiner/gleich/größer 0:
  "this" kleiner/gleich/größer als Objekt o
- Standarddatentypen (z.B. String) implementieren Comparable

ToDo-Listen in BlueJ



Erweiterung von ToDo-Listen (jUML)

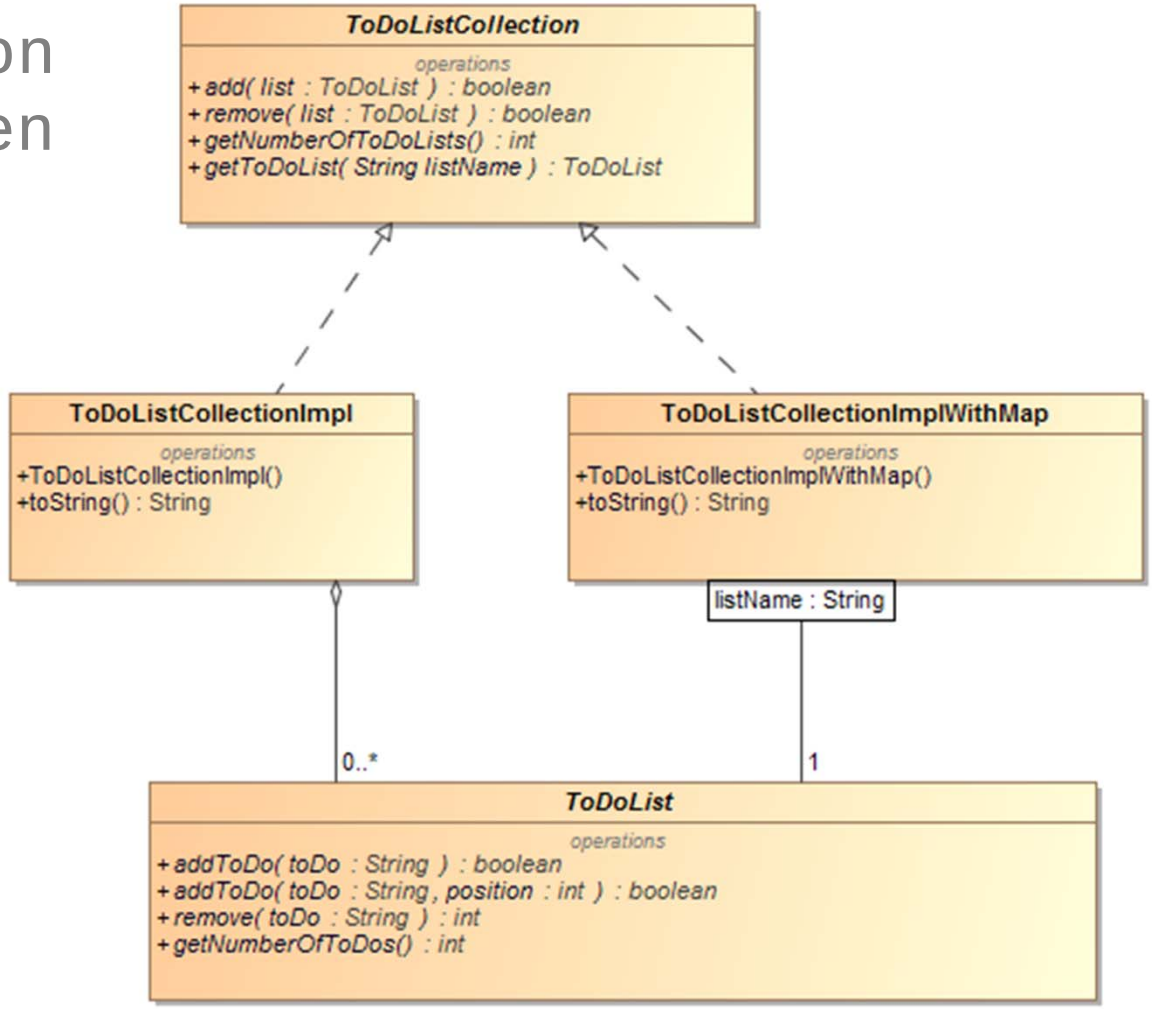
(gegenüber Klausurversion)



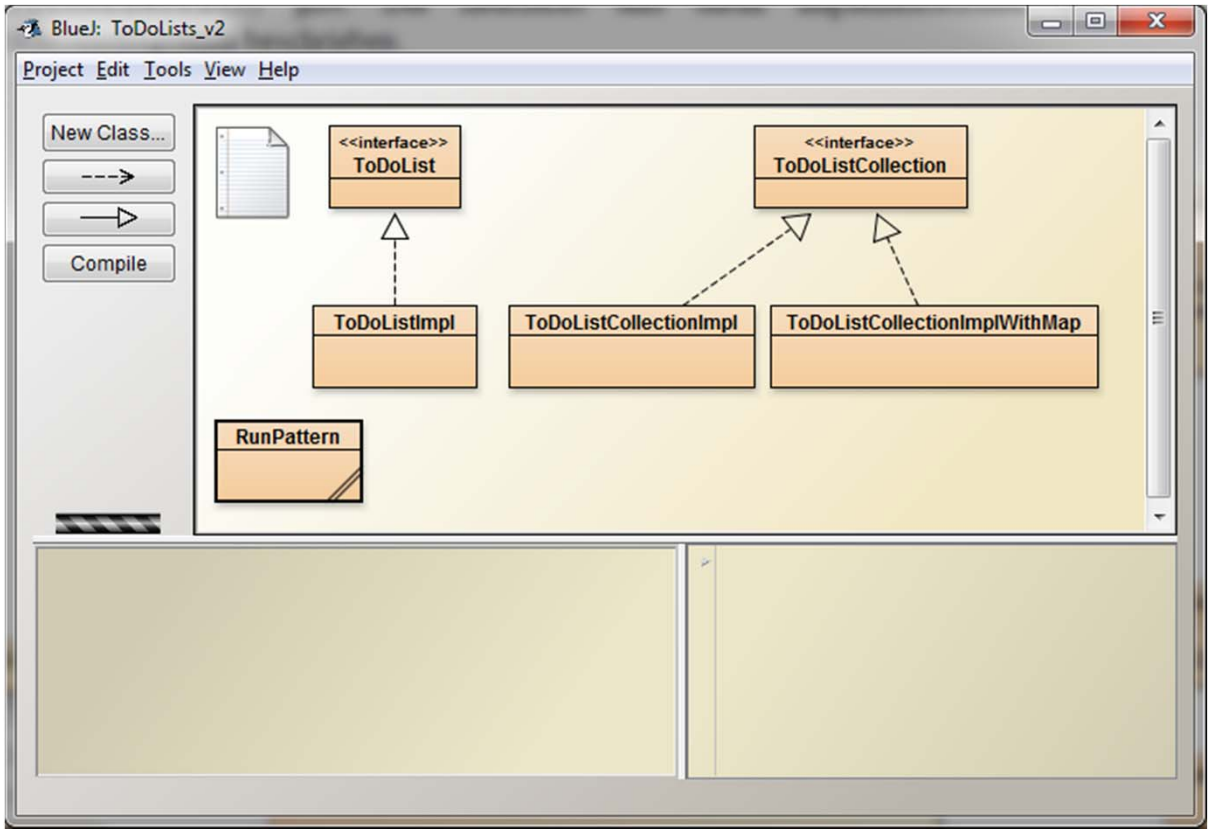
java.util.Map (Auszug)

```
public interface Map<K,V> {
  ...
  public boolean containsKey (Object key);
  public boolean containsValue (Object value);
  public V get (Object key);
  public V put (K key, V value);
  public V remove (Object key);
  public int size();
  public Set<K> keySet();
  public Collection<V> values();
  ...
}
```

ToDoListCollection Implementationen (dUML)



ToDo-Listen erweitert (v2) in BlueJ



Erweiterter Selbsttest

http://bit.do/OOSE_Test

- I. Allg. werden jede Woche zusätzliche Fragen ins Web gestellt.
- Um den Aufwand der Beantwortung „alter“ Fragen zu reduzieren, können ab sofort Fragen/Abschnitte im Test übersprungen werden.
- Bei Angabe einer Emailadresse wird das Ergebnis des Selbsttestes an diese Emailadresse zugeschickt.