HBO Graduaat Informatica Optie Programmeren

Java In Depth

Collection Framework

cvo leerstad

volwassenenonderwijs

Generic Collection Library

- Manieren om objecten te verzamelen in Java
  - **Array**
    - **Fixed** size
  - **java.util library**
    - Container interfaces & classes
    - → **collection**

Introduction

- Collection?
  - Object dat meerdere elementen verzamelt
  - Opslag, opvragen, manipul. & uitwisselen data
  - Items stellen een '**natural group**' voor
- Collections framework?
  - **Architectuur**
  - Alle frameworks bevatten
    - **Interfaces** ➔ abstract data types
    - **Implementations** ➔ reusable data structures
    - **Algorithms** ➔ methods (searching, sorting, …)
      - Polymorphic

Introduction

- Advantages
  - Vermindert **last** van programmeren
  - Verhoogt **snelheid** & **kwaliteit**
  - **Interoperability**
  - Vermindert noodzaak **kennis** verschillende API's
  - Minder last om nieuwe API's te **ontwikkelen**
  - Software **herbruikbaarheid**

Introduction

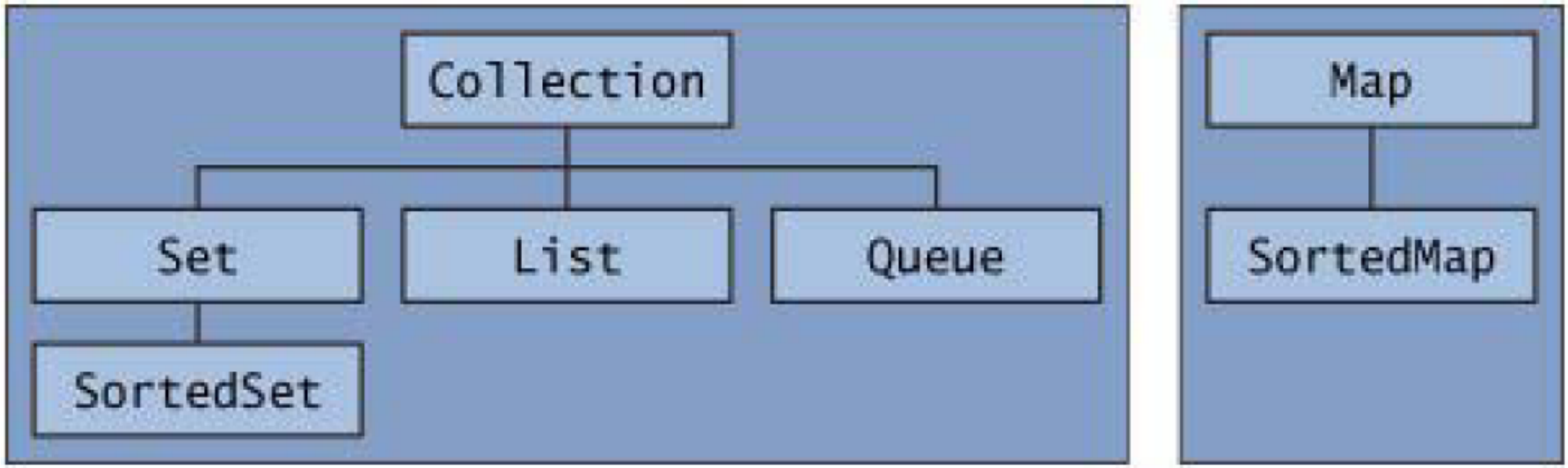
- Disadvantages
  - **Geen kennis** meer v… …lectie (referenties na…
  - **Geen k**… …Collection
  - **P**…

Generics to the rescue!

Collection framework interfaces

- All core collections are generic

```
public interface Collection<E>
```



Collection framework interfaces

- Collection
  - Root of the collection interface
  - Represents group of objects → elements
- Set
  - Cannot contain duplicate elements
- List
  - Ordered collection
  - Can contain duplicate elements
  - Precise control when inserting an element
  - Access elements by integer

Collection framework interfaces

- Queue
- Map
  - Object that maps keys to values
  - Cannot contain duplicate keys
- SortedSet
  - Maintains elements in ascending order
- SortedMap
  - Map analogue of SortedSet

The Collection Interface

- Groepeert objecten (**elementen**)
- **Root** interface ➔ maximale ‘generality’
  - **ALLE** collection implementaties hebben een constructor met als **argument** een ***Collection***
  - Constructor ➔ initialiseren collection met alle elementen uit argument ***Collection***
- **Niet alle** methodes van de Collection interface zijn **geïmplementeerd ➔ UnsupportedOperationException**

The Collection Interface

- Basic Operations
  - size()
  - isEmpty()
  - contains(Object o)
  - add(Object o)
  - remove(Object o)
  - iterator()

3. The Collection Interface

- Bulk Operations
  - containsAll(Collection c)
  - addAll(Collection c)
  - removeAll(Collection c)
  - retainAll(Collection c)
  - clear()
- Array Operations
  - toArray()

The Iterator Interface

- Introduction
  - Iterator pattern

Sequentiële toegang tot elementen van een verzamelObject ZONDER onderliggende voorstelling te onthullen.

The Iterator Interface

- Iterator Design Pattern
  - Om inhoud van een container te bekijken
  - Meerdere ‘traversals’ over collectie
  - Op universele manier (Polymorphic iteration)

4. The Iterator Interface

- Iterator Design Pattern
  - Client behandelt enkel abstracte classes (Aggregate & Iterator)
  - Client vraagt aan aggregate zijn iterator ➔ ConcreteIterator
  - Iterator
    - **first()**
    - **next()**
    - **isDone()**
    - **currentItem()**
  - Iterator Pattern ➔ scheiden aggregate van iterator

4. The Iterator Interface

- java.util.Iterator
  - Interface
  - Uni-directioneel overlopen collectie
  - Methods
    - **hasNext()**
    - **next()**
    - **remove()**
- **Example**

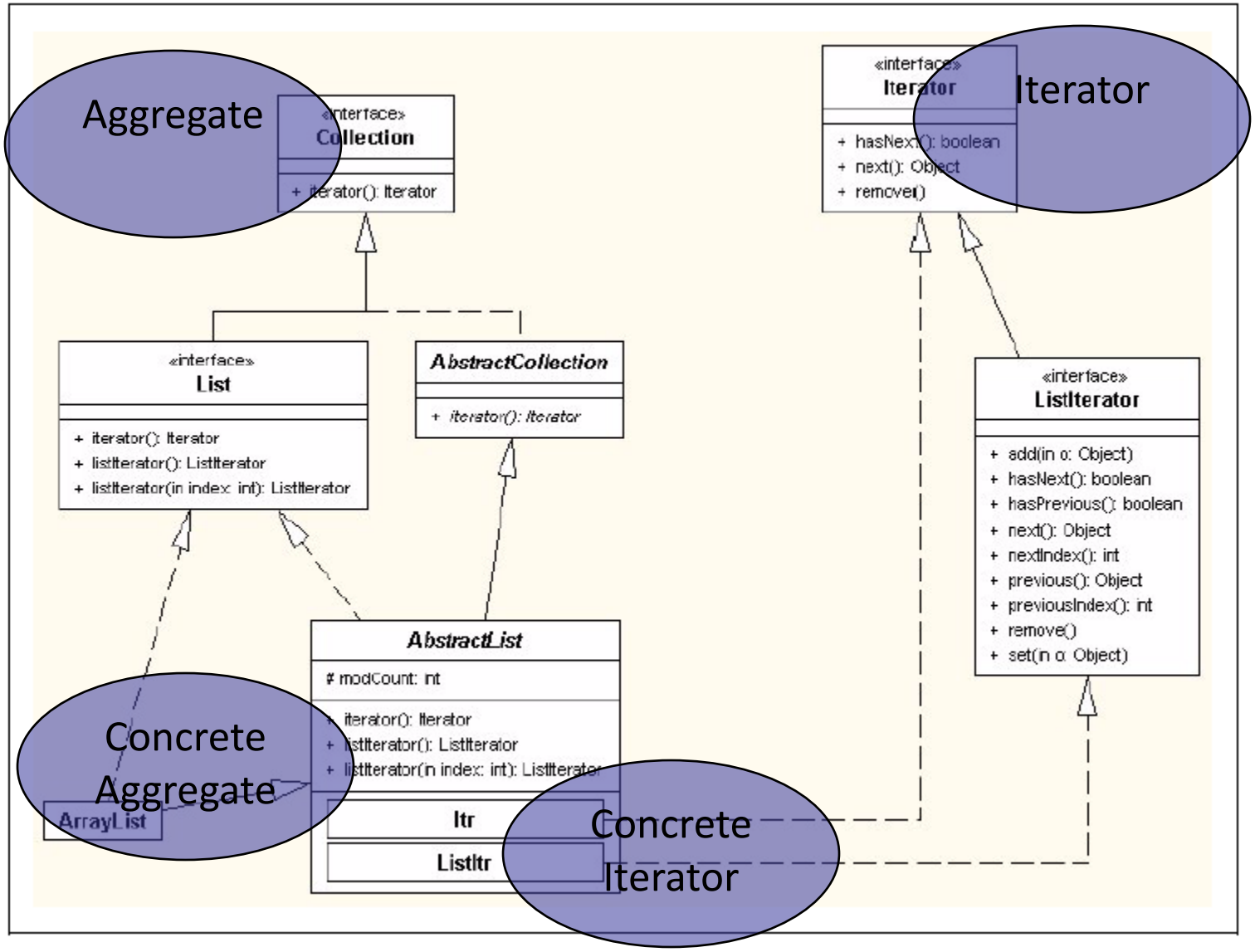
The Iterator Interface

- java.util.Iterator
  - Filteren collectie

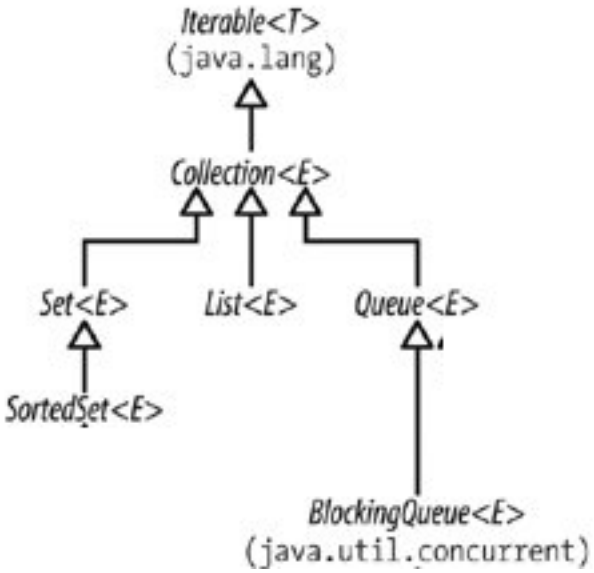
```
static void filter(Collection c){
        for(Iterator it=c.iterator();it.hasNext();){
                if (!cond(it.next())){
                        it.remove();
                }
        }
}
```

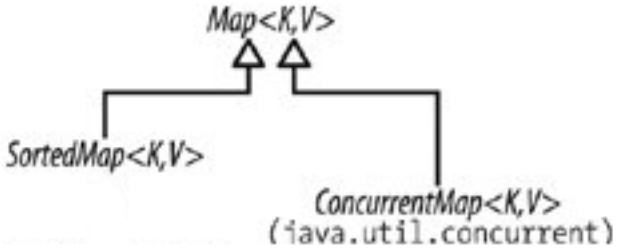
  - Polymorphic algoritme

4. The Iterator Interface



Collection Interfaces





Collection Interfaces

- Basic core collection interfaces
  - Collection : Root
  - List
  - Set
  - Map
  - SortedSet
  - SortedMap

The List Interface

- Geordende collectie
- Sequentie
- Duplikaten toegestaan
- Extra operaties
  - Positional access
  - Search
  - List iteration
  - Range view

The List Interface

- 2 lijst implementaties in JDK
  - ArrayList
  - LinkedList

The List Interface

- Collection operations
  - remove(Object o) ➔ enkel 1ste
  - add() & addAll() ➔ op einde lijst
- Positional access & search operations
  - set(int index, Object e) ➔ oud Object
  - remove(int index)
  - indexOf(Object e), lastIndexOf(Object e)
  - addAll(Collection c)
  - addAll(int index,Collection c)
  - Opm : Arrays.asList ➔ List

The List Interface

- Iteration operation
  - list.iterator() ➔ Iterator
  - list.listIterator() ➔ ListIterator
    - Bi-directioneel, toevoegen, opvragen tijdens iteratie
    - Extra methods
      - hasPrevious(), previous(), nextIndex(), previousIndex(), set(), add()
  - listIterator() ➔ begin lijst
  - listIterator(index i) ➔ vanaf i

The List Interface

- Iteration operation
  - Index lijst :

Elm		Elm0		Elm1	
index	0		1		2

  - Lengte lijst = n ➔ n+1 indexen (0 tem n)
  - !! Eerste previous() na reeks van next()
    ➔ zelfde element als laatste next()
    (en omgekeerd)

The List Interface

- Iteration operation
  - nextIndex() & previousIndex()
    - Om positie te kennen van element
    - Creatie tweede ListIterator vanaf zelfde plaats
    - 1st element ➔ previousIndex() = -1
    - Laatste element ➔ nextIndex() = list.size()

The List Interface

- Range view Operation
  - subList()
  - **View** van originele lijst ➔ **niet**-structurele veranderingen zijn zichtbaar in **lijst & sublijst** !!

The List Interface

- Algorithms
  - Collections ➔ manipuleren lijsten
    - sort(List)
    - shuffle(List)
    - indexOfSubList(List, List)
    - lastIndexOfSubList(List, List)
    - rotate(List, int)
    - swap(List, int, int)
    - replaceAll(List, Object, Object)
    - reverse(List)
    - fill(List, Object)
    - copy(List, List)
    - list(Enumeration)
    - binarySearch(List, Object)

Samenvatting

	sorted	duplicates	key-value	nulls	comment
ArrayList	✗	✓	✗	✓	
LinkedList	✗	✓	✗	✓	

The Set Interface

- Kan **GEEN** duplikaten bevatten
- Extends **Collection**
- **NO** extra methods
- equals() & hashCode()➔ indien sets zelfde elementen hebben
- 3 implementaties
  - HashSet
  - LinkedHashSet
  - TreeSet

The Set Interface

- Basic operations
  - Refereer (altijd) naar interface type ➔ veranderen van implementatie door constructor te veranderen
  - Vb Set s = new HashSet(); ➔ indien gesorteerd : s = new TreeSet();
- Bulk operations
  - Makkelijk voor wiskundige berekeningen (doorsnede, vereniging, …)
- Array operations

Samenvatting

	sorted	duplicates	key-value	nulls	comment
ArrayList	✗	✓	✗	✓	
LinkedList	✗	✓	✗	✓	
TreeSet	✓	✗	✗	✓	
HashSet	✗	✗	✗	✓	

The Map Interface

- Mapping **key – value**
- **GEEN** duplikate keys
- **1** key = **1** value
- 3 implementaties
  - HashMap
  - LinkedHashMap
  - TreeMap
- Collection views voor iteration
- contains() – containsValue()

The Map Interface

- Basic operations
  - Zelfde gedrag als in Hashtable
  - equals() ➔ zelfde key-value mappings
- Bulk operations
  - putAll(Map p) ➔ overschrijven oude key-value
- Collection views
  - keySet() ➔ keys
  - values() ➔ values
  - entrySet() ➔ key-value pairs

The Map Interface

- Collection views - Map Algebra
  - Via views : containsAll(), equals()
- Multimaps
  - **1** key – **multiple** values

Samenvatting

	sorted	duplicates	key-value	nulls	comment
ArrayList	✗	✓	✗	✓	
LinkedList	✗	✓	✗	✓	
TreeSet	✓	✗	✗	✓	
HashSet	✗	✗	✗	✓	
TreeMap	✓	✗	✓	✓	
HashMap	✗	✗	✓	✓	

Queue Interface

```
public interface Queue<E> extends Collection<E> {
    boolean offer(E o);
    E poll();
    E remove();
    E element();
    E peek();
}
```

	Throws exception	Returns special value
Insert	add(E o)	offer(E o)
Remove	remove()	poll()
Examine	element()	peek()

Object Ordering

- Collections.sort(List<T> l)
- Sorteren ➔implements Comparable
  - String : a-z
  - Date : chronologisch
  - Nummers : Integer, Byte, Short, Long, Double, Float, BigInteger, BigDecimal
  - File : a-z pathname
- Indien niet ‘Comparable’ ➔ ClassCastException

The Comparable Interface

- Interface Comparable heeft 1 methode : public int **compareTo**(To);
- Methode 'returnt' int:
  - Negatief : **this** kleiner dan **o**
  - 0 : **this** = **o**
  - Positief : **this** groter dan **o**
- Indien o niet vergelijkbaar is met this → ClassCastException

The Comparator Interface

- Indien sorteren
  - op ‘onnatuurlijke’ wijze
  - Objecten die NIET Comparable zijn
- ➔ Comparator : object die een ‘ordening’ omvat.
- 1 methode :
  int compare(To1,To2)
  - Negatief : **o1** kleiner dan **o2**
  - 0 : **o1** = **o2**
  - Positief : **o1** groter dan **o2**

The SortedSet Interface

- Set die zijn elementen automatisch sorteert (ascending; natural order OF Comparator)
- Extra operaties
  - Range view
    - subSet(T from, Tto)
    - headSet(T to)
    - tailSet(T from)
  - Endpoints
    - first()
    - last()
  - Comparator access
    - comparator()

The SortedSet Interface

- Set operations
  - Inherits from Set
    - iterator()
    - toArray()
- Standard Constructors (Implementaties van SortedSet)
  - Parameter Collection ➔ alle elementen moeten (mutually)Comparable zijn
  - 2 andere :
    - + Comparator ➔ leeg, sortering volgens Comparator
    - + SortedSet ➔ zelfde elm & sortering als parameter

The SortedSet Interface

- Range view operations
  - subSet(Object from, Object to)
  - headSet(Object to)
  - tailSet(Object from)
- Endpoints operations
  - first() ➔ eerste element
  - last() ➔ laatste element
- Comparator accessor
  - comparator() ➔ null indien natural ordening

The SortedMap Interface

- Map die zijn elementen (keys) automatisch sorteert (ascending; natural order OF Comparator)
- Extra operaties
  - Range view
    - subMap(Object fromKey, Object toKey)
    - headMap(Object toKey)
    - tailMap(Object fromKey)
  - Endpoints
    - firstKey()
    - lastKey()
  - Comparator access
    - comparator()

The SortedMap Interface

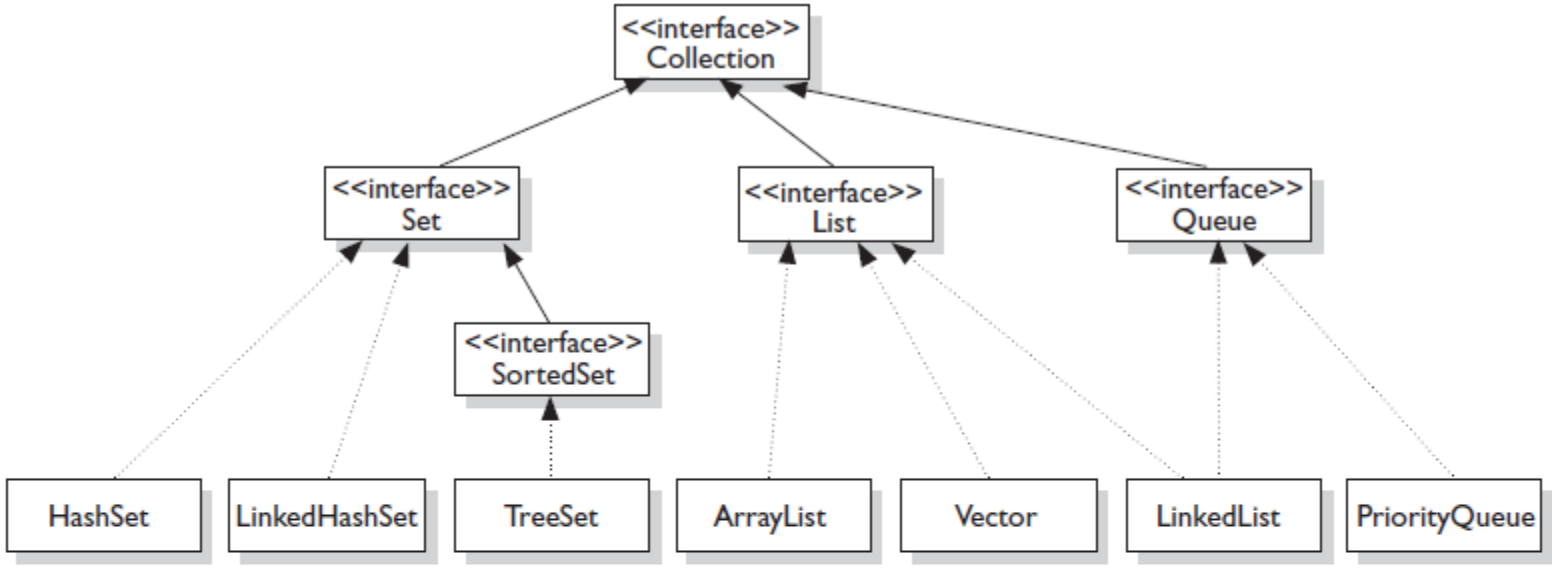
- Map operations
  - Operaties die SortedMap van Map overerft
  - MAAR
    - iterator() op collection views van Map ➔ steeds op volgorde keys !!
    - toArray() op collection views van Map ➔ steeds op volgorde keys !!

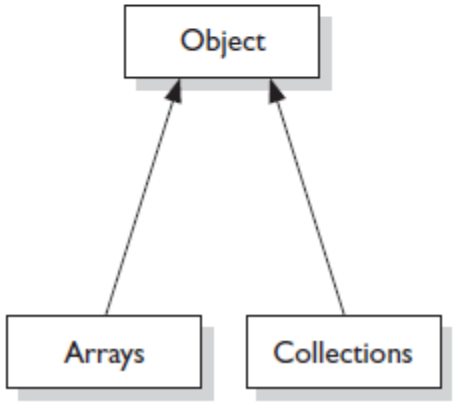
The SortedMap Interface

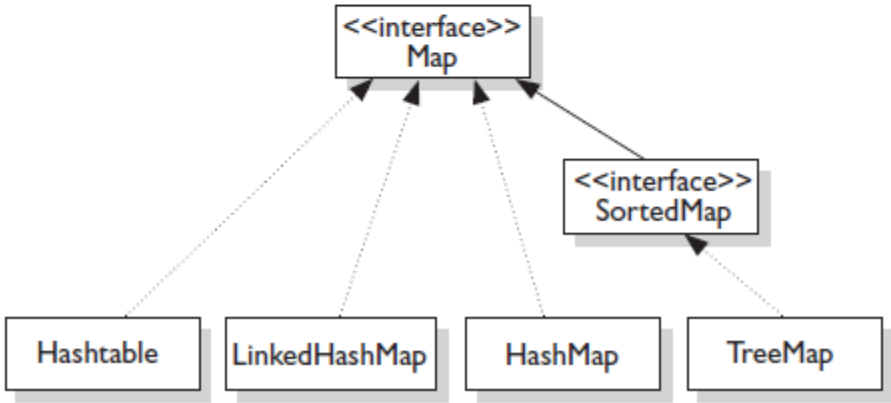
- Standard Constructors (Implementaties van SortedMap)
  - Parameter Map ➔ natural ordening Keys
  - 2 andere :
    - + Comparator ➔ leeg, sortering volgens Comparator
    - + SortedMap ➔ zelfde mappings & sortering als parameter (zie ook +Map)
- Comparison to SortedSet
  - Analoog aan SortedSet (mits kleine veranderingen)

The Collections Class

- java.util.Collections
- Statische methodes ➔ interfaces
  - Synchronisatie (thread safe)
  - Read-only versions
  - Sortering
  - Meerdere kopijen van een Object
  - Collection/Map met 1 Object
  - EMPTY_SET, EMPTY_MAP, EMPTY_LIST
  - …


<<interface>>
Collection
<<interface>>
Set
<<interface>>
List
<<interface>>
Queue
<<interface>>
SortedSet
HashSet
LinkedHashSet
TreeSet
ArrayList
Vector
LinkedList
PriorityQueue


Object
Arrays
Collections


<<interface>>
Map
<<interface>>
SortedMap
Hashtable
LinkedHashMap
HashMap
TreeMap

Questions ??