Collections

CPSC 1181 – O.O.

Jeremy Hilliker

Summer 2017

Langara.

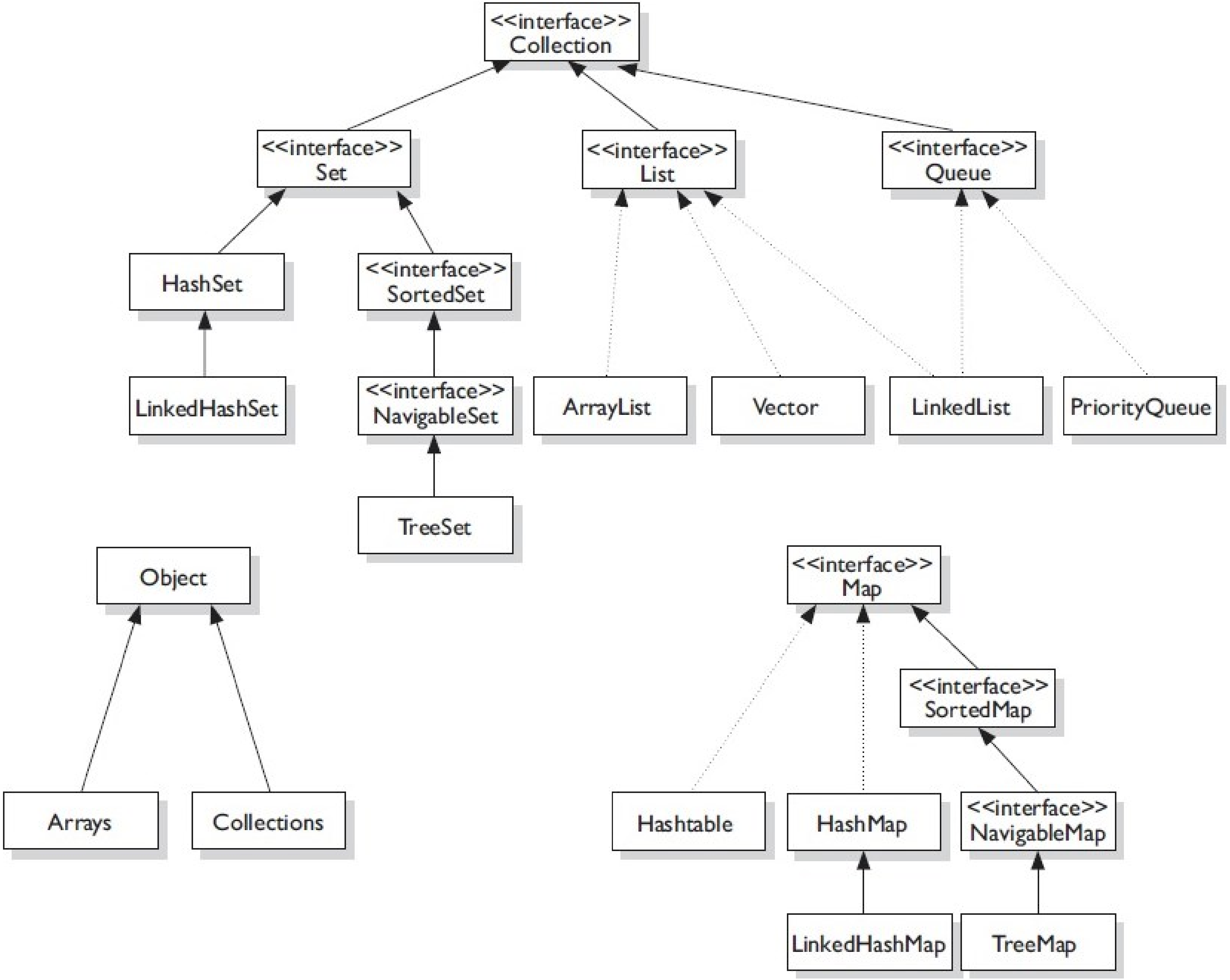
THE COLLEGE OF HIGHER LEARNING.

Outline

- Collections
- ArrayList
- HashSet
- HashMap
- Iterator

Collections

- A groups of common
    - Data structures
    - Abstractions
    - Algorithms
- Benefits
    - Reduced effort
        - To program your own things
        - To learn new APIs (they will use collections)
        - To design APIs (use collections)
    - Increased speed and quality
    - Interoperability between APIs
    - Foster reuse


<<interface>> Collection
<<interface>> Set
<<interface>> List
<<interface>> Queue
HashSet
<<interface>> SortedSet
LinkedHashSet
<<interface>> NavigableSet
TreeSet
ArrayList
Vector
LinkedList
PriorityQueue
Object
Arrays
Collections
<<interface>> Map
<<interface>> SortedMap
Hashtable
HashMap
<<interface>> NavigableMap
LinkedHashMap
TreeMap

List

- ArrayList

- An ordered sequence of elements
- Can contain duplicates
- Concept of “where” and “position”

- Runtime: some are constant, some are linear

Set

- HashSet

- An unordered* bag of things
- No duplicates

- Runtime: operations run in constant time

Map

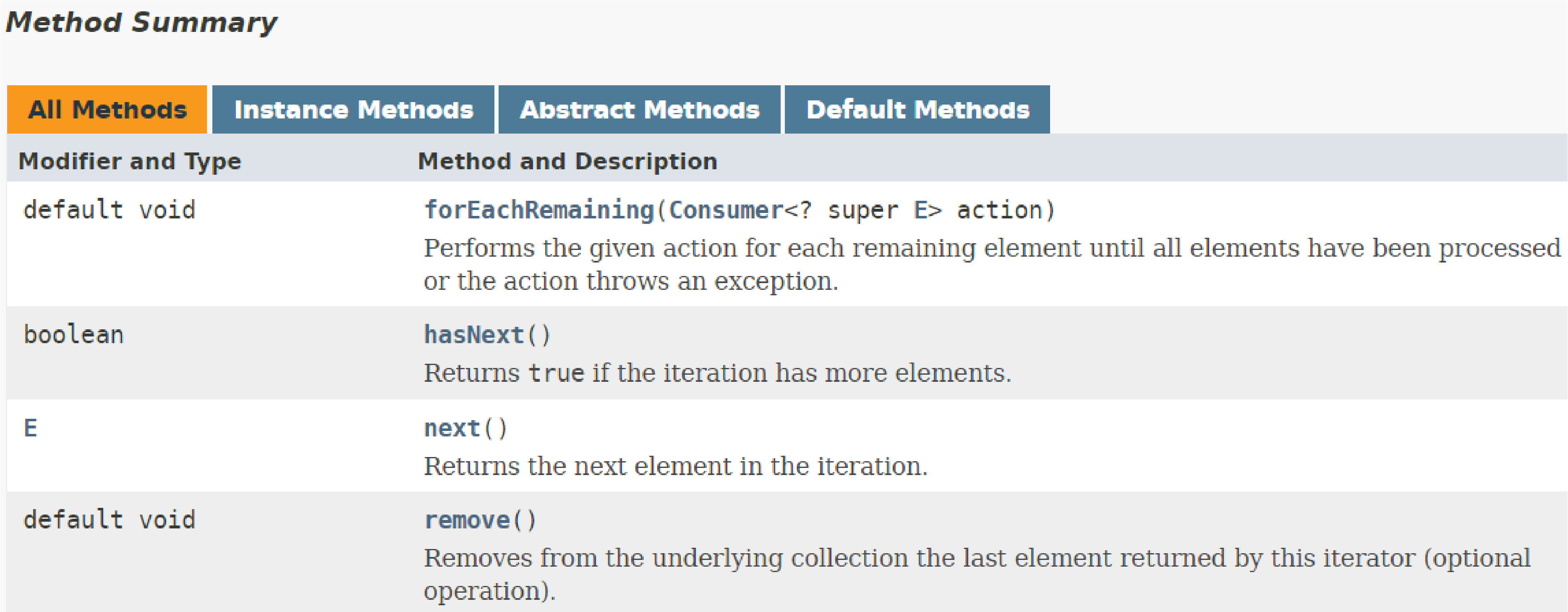
- HashMap

- A set of key-value pairs
- Maps keys to values
- No duplicate keys
- Key associated with one value

- Runtime:
  - most constant
  - Others “amortized” constant time

Iterator

- Visits every element in a collection
- Optional: can remove last visited



Method Summary

Modifier and Type	Method and Description
`default void`	`forEachRemaining(Consumer<? super E> action)` Performs the given action for each remaining element until all elements have been processed or the action throws an exception.
`boolean`	`hasNext()` Returns `true` if the iteration has more elements.
`E`	`next()` Returns the next element in the iteration.
`default void`	`remove()` Removes from the underlying collection the last element returned by this iterator (optional operation).

Collections

- sort (fast & stable)
- shuffle
- reverse, fill, copy, swap, addAll
- binarySearch
- composition (frequency, disjoint)
- min , max

Arrays

- binarySearch
- copyOf
- equals
- fill
- hashCode
- accumulate
- sort
- setAll
- toString