JAC444 - BTP400 Course Object-Oriented Software Development II - Java

Collections

Segment 6

Algorithms

The Java Collections Framework

- A collection (container) is an object that groups multiple elements into a single unit

- Operations with a container
  - Put an object in
  - Take an object out
  - Iterate over everything in the container (sometimes with condition)
  - Create a container with modified elements from an initial container
  - Information about a specific object
    - Is it in the container
    - How many objects of this type are in container
    - Is an equivalent object in the container

Interfaces and Classes

- Java 2 had `Vector`, `Hashtable` and `Enumaration`
- Java 8 has Interfaces, Implementations, and Algorithms
- Core Interfaces
  - `Set`
  - `List`
  - `Map`
  - `Queue`
  - `Deque`
  - `SortedSet`
  - `Sorted Map`

Utility Interfaces

- `Comparator`
- `Iterator`

Utility Classes

- `Collections`
- `Arrays`

The `Collections` Utility Class

- `Collections` class has only static methods
- Most methods operate on `List`
- Example: `public static <T> void sort(List<T> list)`

```
public static <T extends Comparable<? super T>>
void sort(List<T> list)
```

Sorts the specified list into ascending order, according to the natural ordering of its elements.

All elements in the list must implement the Comparable interface

Sort Example

Sort the command line arguments in lexicographically (alphabetical order)

```
import java.util.*;

public class SortExample {

  public static void main( String args[] ) {

    List list = new ArrayList();

    for ( int i = 0; i < args.length; i++ )
      list.add( args[i] );

    Collections.sort( list );

    System.out.println( list );
  }
}
```

The `Array` Utility Class

- The `Array` utility class contains various methods for manipulating arrays (such as sorting and searching).

For example:

```
public static <T> List<T> asList(T array)
```

returns a fixed-size list from the specified array

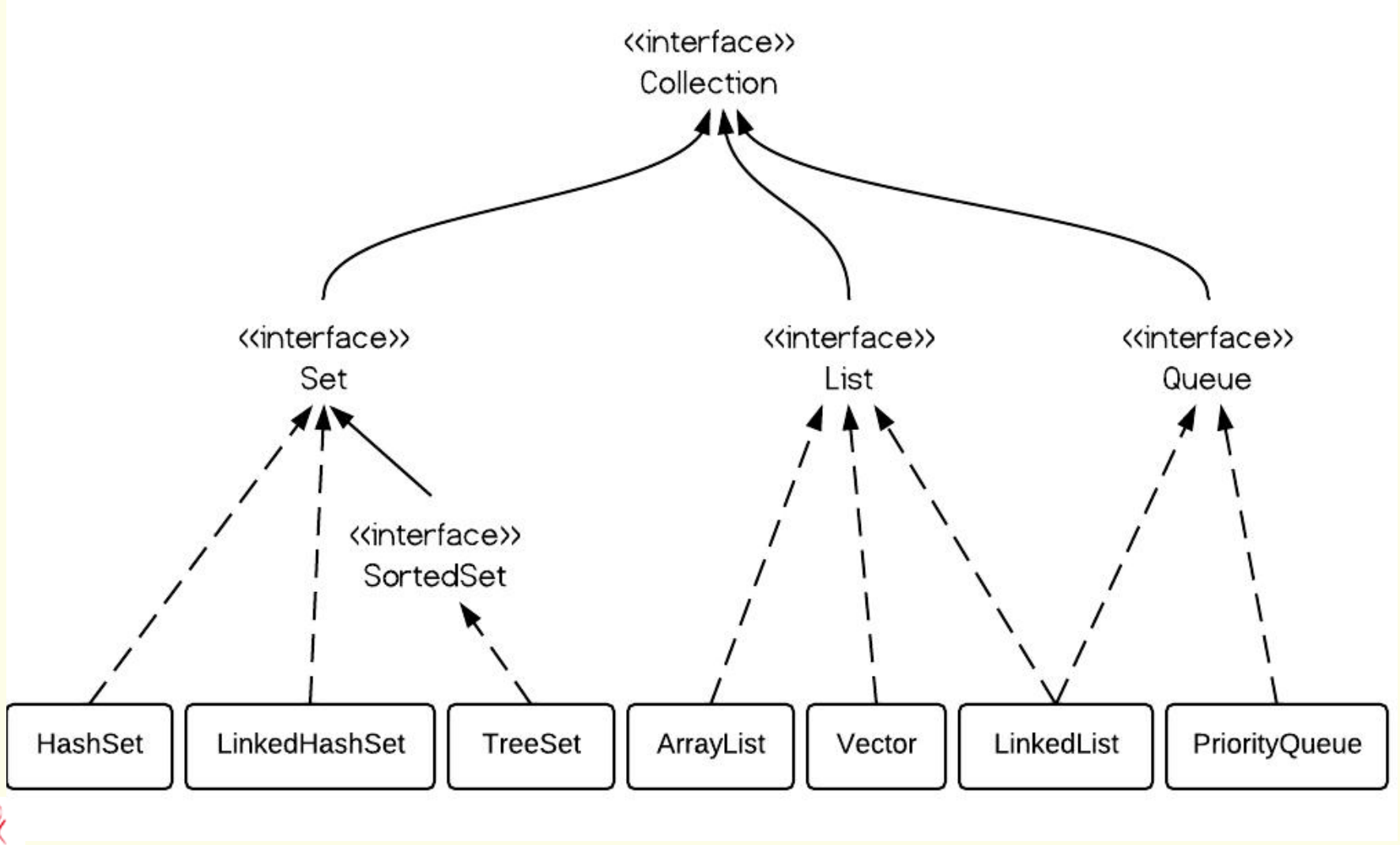
```
List<String> students =
Arrays.asList("Larry", "John", "Marry");
```

Sort Example using `Arrays`

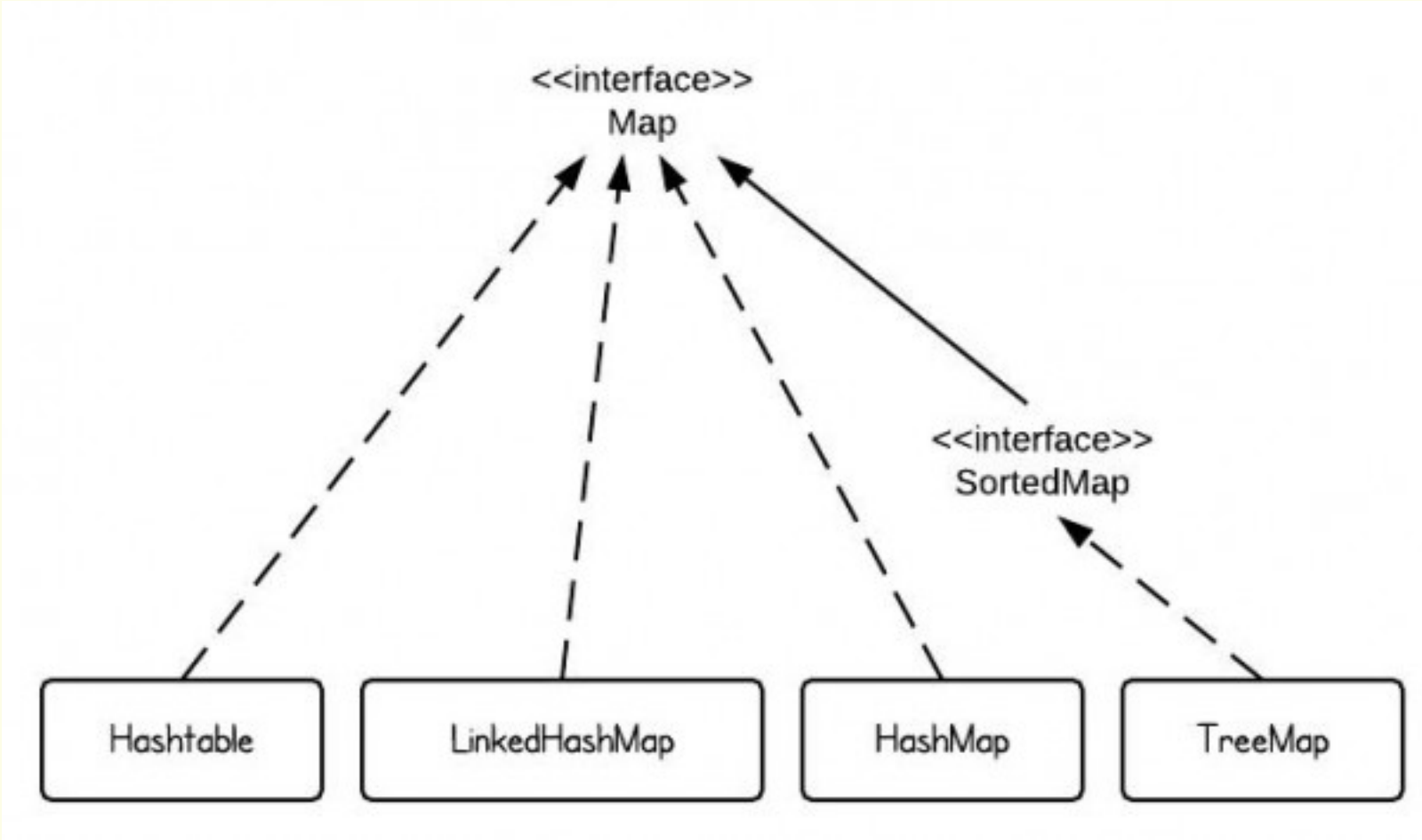
Sort the command line arguments in lexicographically (alphabetical order)

```
import java.util.*;

public class SortExample {

  public static void main( String[] args ) {

    Arrays.sort( args );

    List list = Arrays.asList( args );

    System.out.println( list );
  }
}
```

Hierarchy of `Collection`



Hierarchy of `Map`



Summary of Classes

Interfaces	Hash table	Resizable array	Tree	Linked list	Hash table + Linked list
Set	HashSet		TreeSet		LinkedHashSet
List		ArrayList		LinkedList	
Queue					
Map	HashMap		TreeMap		LinkedHashMap