# Collections

(http://docs.oracle.com/javase/tutorial/collections/ index.html)

## Objectives

- Collections Framework (package java.util):
- List: ArrayList, Vector → Duplicates are agreed
- Set: HashSet, TreeSet → Duplicates are not agreed
- Map: HashMap, TreeMap

# The Collections Framework

- The Java 2 platform includes a new *collections framework*.
- A *collection* is an object that represents a group of objects.
- The Collections Framework is a unified architecture for representing and manipulating collections.
- The collections framework as a whole is not threadsafe.

# The Collections Framework…

- **Reduces programming effort** by providing useful data structures and algorithms so you don't have to write them yourself.
- **Increases performance** by providing high-performance implementations of useful data structures and algorithms.
- **Provides interoperability between unrelated APIs** by establishing a common language to pass collections back and forth.
- **Reduces the effort required to learn APIs** by eliminating the need to learn multiple ad hoc collection APIs.
- **Reduces the effort required to design and implement APIs** by eliminating the need to produce ad hoc collections APIs.
- **Fosters software reuse** by providing a standard interface for collections and algorithms to manipulate them.

# Collection Interfaces

- java.lang.**Iterable**<T>
  - java.util.**Collection**<E>
    - java.util.**List**<E>
    - java.util.**Queue**<E>
      - java.util.**Deque**<E>
    - java.util.**Set**<E>
      - java.util.**SortedSet**<E>
        - java.util.**NavigableSet**<E>
- java.util.**Map**<K,V>
  - java.util.**SortedMap**<K,V>
    - java.util.**NavigableMap**<K,V>

Methods declared in these interfaces can work on a list containing elements which belong to arbitrary type. T: type, E: Element, K: Key, V: Value

Details of this will be introduced in the topic Generic

**3 types of group:**
**List** can contain duplicate elements
**Set** can contain distinct elements only
**Map** can contain pairs <key, value>. Key of element is data for fast searching

**Queue, Deque** contains methods of restricted list.

Common methods on group are: Add, Remove, Search, Clear,…

| Method | Description |
|---|---|
| `add(Object x)` | Adds x to this collection |
| `addAll(Collection c)` | Adds every element of c to this collection |
| `clear()` | Removes every element from this collection |
| `contains(Object x)` | Returns true if this collection contains x |
| `containsAll(Collection c)` | Returns true if this collection contains every element of c |
| `isEmpty()` | Returns true if this collection contains no elements |
| `iterator()` | Returns an Iterator over this collection (see below) |
| `remove(Object x)` | Removes x from this collection |
| `removeAll(Collection c)` | Removes every element in c from this collection |
| `retainAll(Collection c)` | Removes from this collection every element that is not in c |
| `size()` | Returns the number of elements in this collection |
| `toArray()` | Returns an array containing the elements in this collection |

Elements can be stored using some ways such as an array, a tree, a hash table. Sometimes, we want to traverse elements as a list → We need a list of references → Iterator