Programming Concepts Using Java

Quiz 2 Revision

W06:L01: Indirection

Consider two separate implementations:

```
public class CircularArrayQueue<E> {
    public void add (E element){...};
    public E remove(){...};
    public int size(){...};
    ...
}

public class LinkedListQueue<E> {
    public void add (E element){...};
    public E remove(){...};
    public int size(){...};
    ...
}
```

W06:L01: Indirection

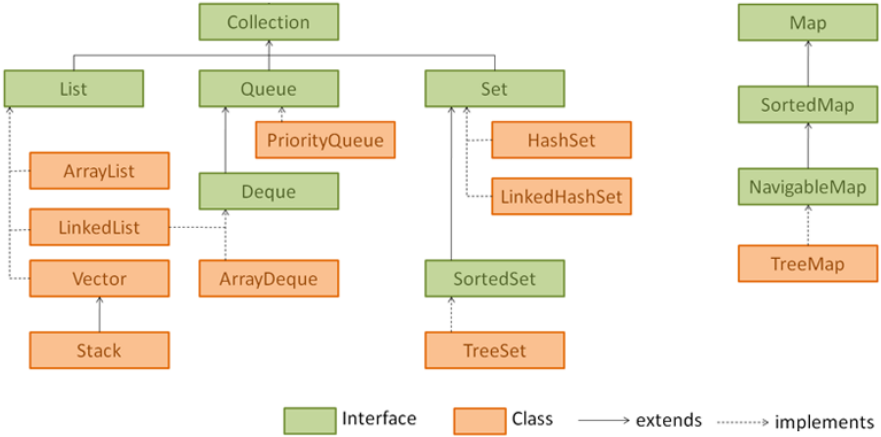
Adding indirection using an interface.

```
interface Queue<E> {
    public void add (E element){...};
    public E remove(){...};
    public int size(){...};
    ...
}
class CircularArrayQueue<E> implements Queue<E>{
    ...
}
class LinkedListQueue<E> implements Queue<E>{
    ...
}
```

W06:L02: Java-Collections

- Collections Framework?
  - collection of interfaces and classes.
  - organizing a group of heterogeneous objects efficiently.
  - All collection classes in Java are declared generic.
  - Framework has several useful classes which have tons of useful methods which makes a programmer task super easy.
  - Some collections allow duplicate elements, while others do not.
  - Some collections are ordered and others are not.
  - Each different type of collection organize their elements in its own way.
  - Reduced development effort by using core collection classes rather than implementing our own collection classes.

W06:L02: Java-Collections



Source: https://www.sneppets.com/java/collections-framework/collection-and-collections-framework-in-java/

W06:L03: Java-Concrete-Collections

- The `List` interface:
  - A list is an ordered collection of objects.
  - A list can have duplicate elements. and we can store multiple null values in a list.
  - The List interface inherits the `Collection` interface, and it adds methods to support access to its elements using indexes.
  - We can add an element at the end of the `List` or at any position identified by an integer index.
- ```
  public interface List<E> extends Collection<E>{
      void add(int index, E element);
      void remove(int index);
      E get(int index);
      E set(int index, E element);
  }
  ```
- List interface implemented classes.
  - `ArrayList`
  - `LinkedList`
  ```

W06:L03: Java-Concrete-Collections

- The `Set` interface
  - A set is a collection without duplicates.
- The following classes implement `Set` interface.
  - `HashSet`: It does not guarantee the ordering of elements during iteration.
    - It stores the elements in a hash table.
    - It uses the hash code of the object being inserted.
    - You can add only one null element to `HashSet`.
  - `LinkedHashSet`: It keeps the element order as the elements were inserted.
    - It has a combined implementation of hash table and linked list.
    - You can add only one null element to `LinkedHashSet`.
  - `TreeSet`: It implements `SortedSet` which is sub interface of `Set`.
    - It uses tree data structure to store values.
    - Elements in `TreeSet` are sorted by natural ascending order. `Null` value is not allowed.

W06:L03: Java-Concrete-Collections

- The `Queue` interface.
  - A queue is a collection of objects on which operations can only be performed at two ends of the queue.
  - A queue has two ends known as head and tail.
  - In the simple queue, objects are added to the tail and removed from the head and the object added first will be removed first.
- `Queue` interface implemented classes.
  - `PriorityQueue`
  - `ArrayDeque`

W06:L04: Maps

- The `Map` interface
  - Key-value structures come under the Map interface.
  - Two type parameters
  - `K` is the type for keys
  - `V` is the type for values
- `Map` interface implemented classes.
  - `HashMap`
  - `TreeMap`
  - `LinkeHashMap`