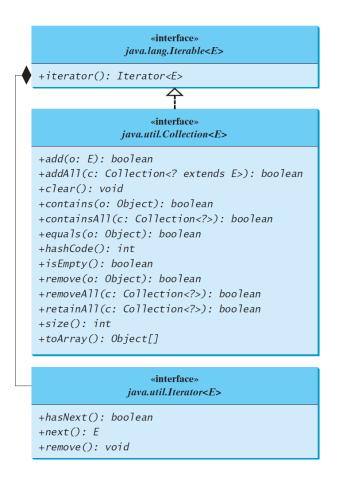# 23-813-0207: Lab 4 - Java Programming Lab

Lab exercise - 5

Date: 20/3/2024

1. Write a program demonstrating methods in Java's *java.util.Collection* Interface using a *HashSet* object.
2. Write a program demonstrating Java's *java.util.List* Interface methods using an *ArrayList* object. Also, print the elements of the ArrayList using a *java.util.ListIterator.*
3. Demonstrate the working of stack data structure using *java.util.Stack* class.
4. Demonstrate the methods in *java.util.Queue* Interface using a *LinkedList* class.
5. Demonstrate the working of a Map data structure using *HashMap* object.

**Refer to the UML diagram given below

«interface»
java.lang.Iterable<E>

```
+iterator(): Iterator<E>
```

«interface»
java.util.Collection<E>

```
+add(o: E): boolean
+addAll(c: Collection<? extends E>): boolean
+clear(): void
+contains(o: Object): boolean
+containsAll(c: Collection<?>): boolean
+equals(o: Object): boolean
+hashCode(): int
+isEmpty(): boolean
+remove(o: Object): boolean
+removeAll(c: Collection<?>): boolean
+retainAll(c: Collection<?>): boolean
+size(): int
+toArray(): Object[]
```

«interface»
java.util.Iterator<E>

```
+hasNext(): boolean
+next(): E
+remove(): void
```

«interface»
java.util.Collection<E>

«interface»
java.util.List<E>

```
+add(index: int, element: Object): boolean
+addAll(index: int, c: Collection<? extends E>)
 : boolean
+get(index: int): E
+indexOf(element: Object): int
+lastIndexOf(element: Object): int
+listIterator(): ListIterator<E>
+listIterator(startIndex: int): ListIterator<E>
+remove(index: int): E
+set(index: int, element: Object): Object
+subList(fromIndex: int, toIndex: int): List<E>
```

```
                  «interface»
              java.util.Collection<E>
```

```
                  «interface»
              java.util.Queue<E>

  +offer(element: E): boolean
  +poll(): E

  +remove(): E

  +peek(): E

  +element(): E
```

```
              java.util.Vector<E>
```

```
              java.util.Stack<E>

  +Stack()
  +empty(): boolean
  +peek(): E
  +pop(): E
  +push(o: E): E
  +search(o: Object): int
```

```
                  «interface»
              java.util.Map<K,V>

  +clear(): void
  +containsKey(key: Object): boolean

  +containsValue(value: Object): boolean

  +entrySet(): Set<Map.Entry<K,V>>
  +get(key: Object): V
  +isEmpty(): boolean
  +keySet(): Set<K>
  +put(key: K, value: V): V
  +putAll(m: Map<? extends K,? extends
   V>): void
  +remove(key: Object): V
  +size(): int
  +values(): Collection<V>
```

```
                  «interface»
              java.util.Iterator<E>
```

```
                  «interface»
              java.util.ListIterator<E>

  +add(element: E): void
  +hasPrevious(): boolean

  +nextIndex(): int
  +previous(): E
  +previousIndex(): int
  +set(element: E): void
```