**VECTEUR → ArrayList**

**ArrayList()**

**ArrayList(int initialCapacity)**

**int size()**

**boolean isEmpty()**

**E get(int index)**

**E set(int index, E e)**

**boolean add(E e)**

**void add(int index, E e)**

**E remove(int index)**

**boolean remove(Object o)**

**boolean contains(Object o)**

**int indexOf(Object o)**

**String toString()**

**Iterator<E> iterator()**

**PILE → ArrayDeque**

**ArrayDeque()**

**ArrayDeque(int numElements)**

**int size()**

**boolean isEmpty()**

**E peek()**

**void push(E e)**

**E pop()**

**String toString()**

**FILE → ArrayDeque**

**ArrayDeque()**

**ArrayDeque(int numElements)**

**int size()**

**boolean isEmpty()**

**E getFirst()**

**void addLast(E e)**

**E removeFirst()**

**String toString()**

**DEQUE → ArrayDeque**

**ArrayDeque()**

**ArrayDeque(int numElements)**

**int size()**

**boolean isEmpty()**

**E getFirst()**

**E getLast()**

**void addFirst(E e)**

**void addLast(E e)**

**E removeFirst ()**

**E removeLast ()**

**String toString()**

**ENSEMBLE → HashSet**

**HashSet()**

**HashSet(int initialCapacity)**

**int size()**

**boolean isEmpty()**

**boolean contains(Object o)**

**boolean add(E e)**

**boolean remove(Object o)**

**String toString()**

**Iterator<E> iterator()**

**DICTIONNAIRE NON TRIE → HashMap**

**HashMap()**

**HashMap(int initialCapacity)**

**int size()**

**boolean isEmpty()**

**boolean containsKey(Object o)**

**V get(Object key)**

**V put(K key,V value)**

**V remove(Object key)**

**Collection<V> values()**

**Set<K> keySet()**

**String toString()**

**LISTE → LinkedList**

**LinkedList()**

**int size()**

**boolean isEmpty()**

**E getFirst()**

**E getLast()**

**E get(int index)**

**E set(int index, E element)**

**void addFirst(E e)**

**void addLast(E e)**

**E removeFirst()**

**E removeLast()**

**void add(int index, E e)**

**boolean remove(Object o)**

**Object remove(int index)**

**boolean contains(Object o)**

**int indexOf(Object o)**

**String toString()**

**Iterator<E> iterator()**

**LISTE SD → LinkedHashSet**

**LinkedHashSet()**

**int size()**

**boolean isEmpty()**

**boolean add(E e)**

**boolean remove(Object o)**

**boolean contains(Object o)**

**String toString()**

**Iterator<E> iterator()**

**ENSEMBLE TRIE → TreeSet**

**TreeSet()**

**int size()**

**boolean isEmpty()**

**boolean contains(Object o)**

**boolean add(E e)**

**boolean remove(Object o)**

**String toString()**

**Iterator<E> iterator()**