CS 2050
Computer Science II

Thyago Mota

METROPOLITAN STATE UNIVERSITY℠
OF DENVER
LIVES TRANSFORMED

Agenda

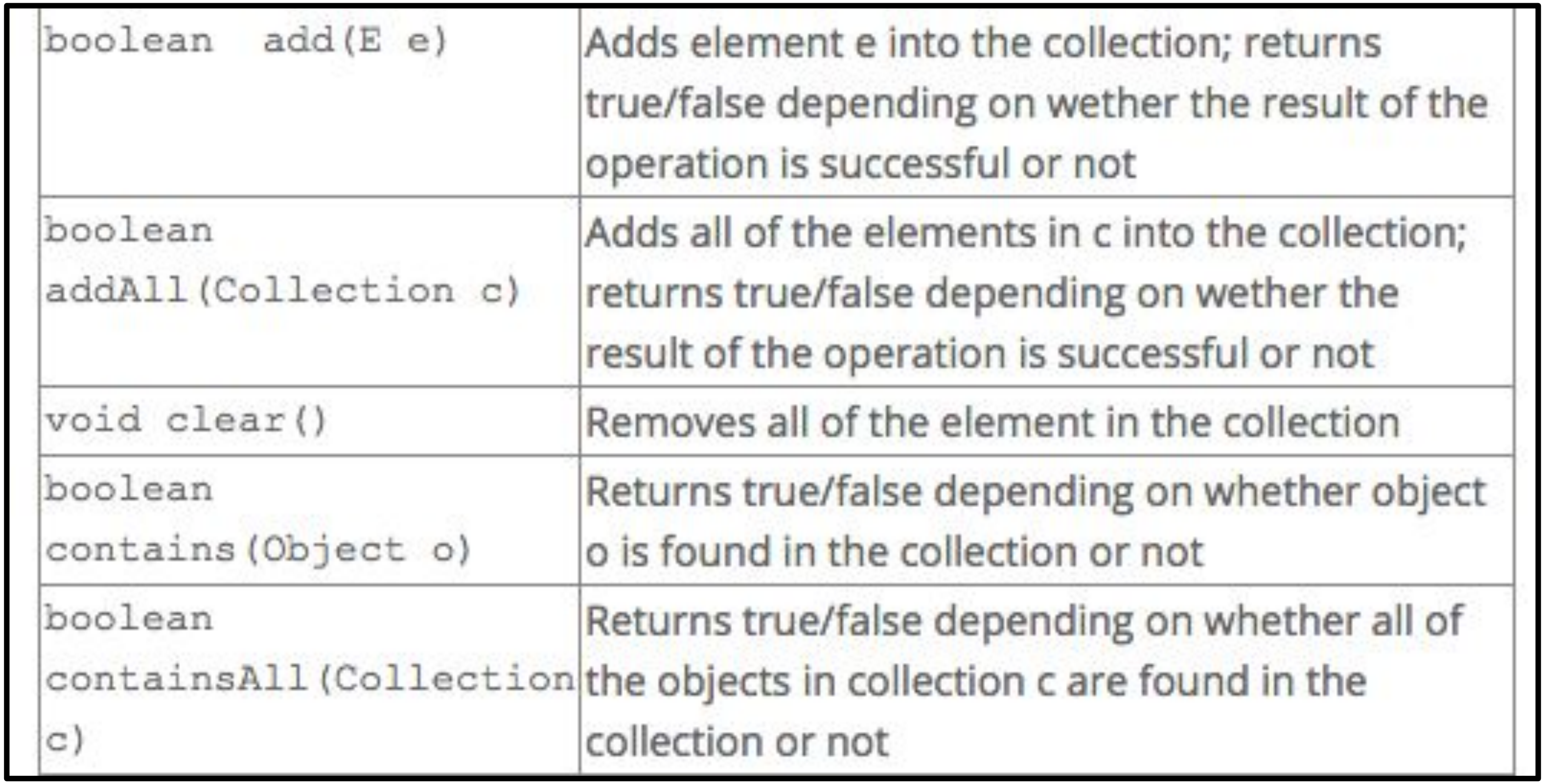
- Overview
- The `Collection` Interface
  - The `List` Interface
  - The `Set` Interface
  - The `Map` Interface
- Summary
- Practice

Overview

- The **Collections Framework** is a Java library for working with groups of individual objects
- `Collection` is the name of the framework's main interface
- It defines methods that are common between all collection types

The Collection Interface

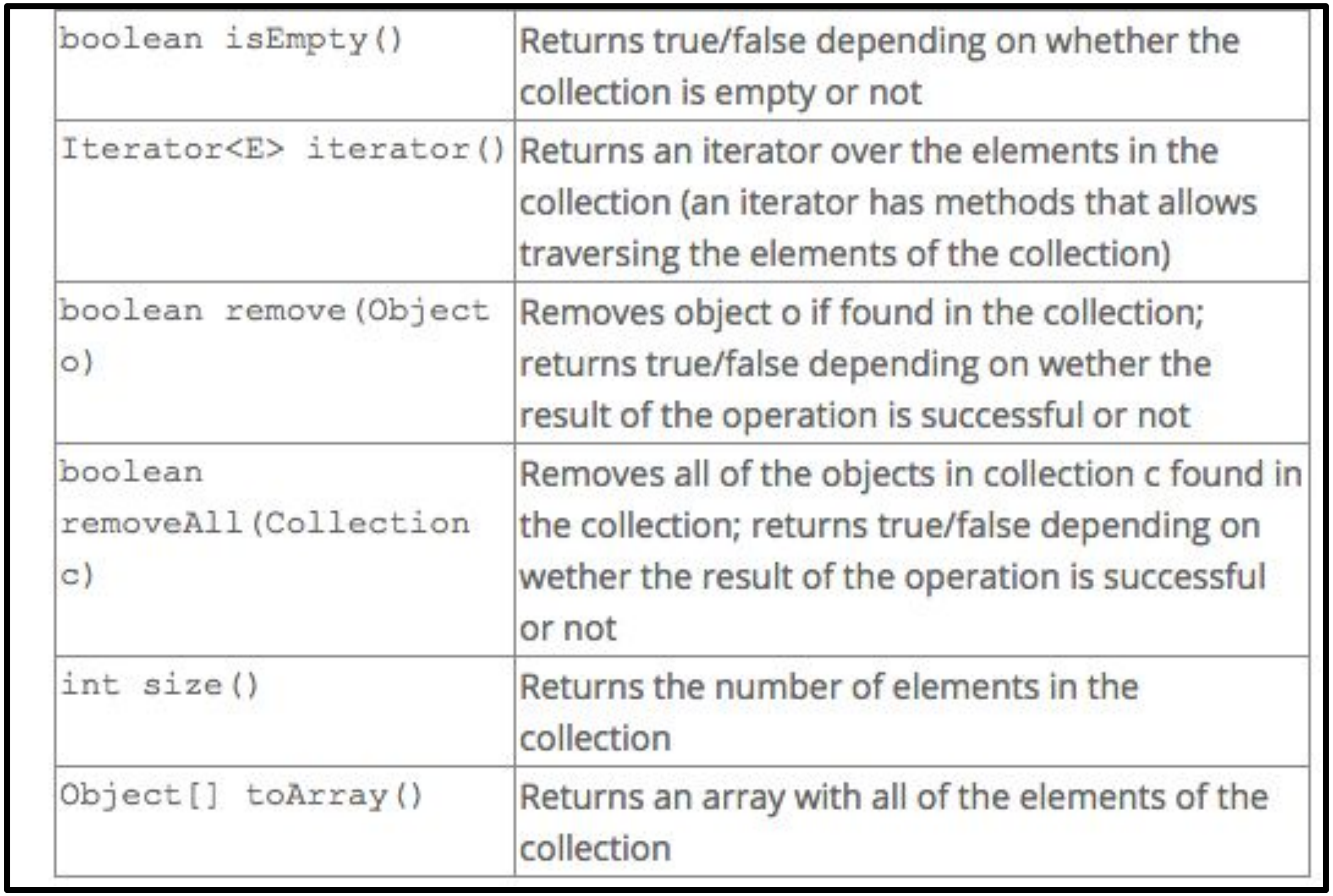
METHODS



`boolean add(E e)`	Adds element e into the collection; returns true/false depending on wether the result of the operation is successful or not
`boolean addAll(Collection c)`	Adds all of the elements in c into the collection; returns true/false depending on wether the result of the operation is successful or not
`void clear()`	Removes all of the element in the collection
`boolean contains(Object o)`	Returns true/false depending on whether object o is found in the collection or not
`boolean containsAll(Collection c)`	Returns true/false depending on whether all of the objects in collection c are found in the collection or not

The Collection Interface

M
E
T
H
O
D
S



`boolean isEmpty()`	Returns true/false depending on whether the collection is empty or not
`Iterator<E> iterator()`	Returns an iterator over the elements in the collection (an iterator has methods that allows traversing the elements of the collection)
`boolean remove(Object o)`	Removes object o if found in the collection; returns true/false depending on wether the result of the operation is successful or not
`boolean removeAll(Collection c)`	Removes all of the objects in collection c found in the collection; returns true/false depending on wether the result of the operation is successful or not
`int size()`	Returns the number of elements in the collection
`Object[] toArray()`	Returns an array with all of the elements of the collection

The Collection Interface

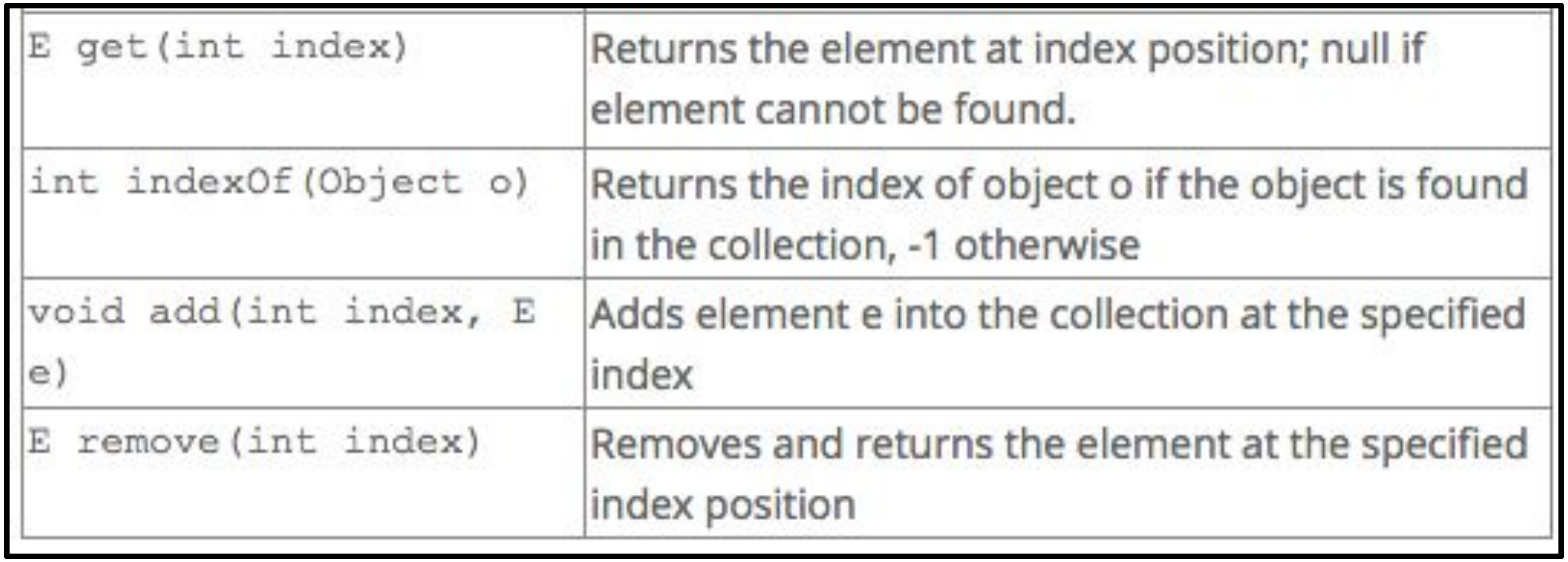
- Specialized Interfaces:
  - `List`
  - `Set`
  - `Map`

The List Interface

- Enables the definition of an ordered collection of objects of the same type

The List Interface

METHODS



`E get(int index)`	Returns the element at index position; null if element cannot be found.
`int indexOf(Object o)`	Returns the index of object o if the object is found in the collection, -1 otherwise
`void add(int index, E e)`	Adds element e into the collection at the specified index
`E remove(int index)`	Removes and returns the element at the specified index position

The List Interface

- Implementations:
    - `ArrayList`
    - `LinkedList`
    - `Stack`

The Set Interface

- Enables the definition of an unordered collection of objects where duplicates are not allowed

The Set Interface

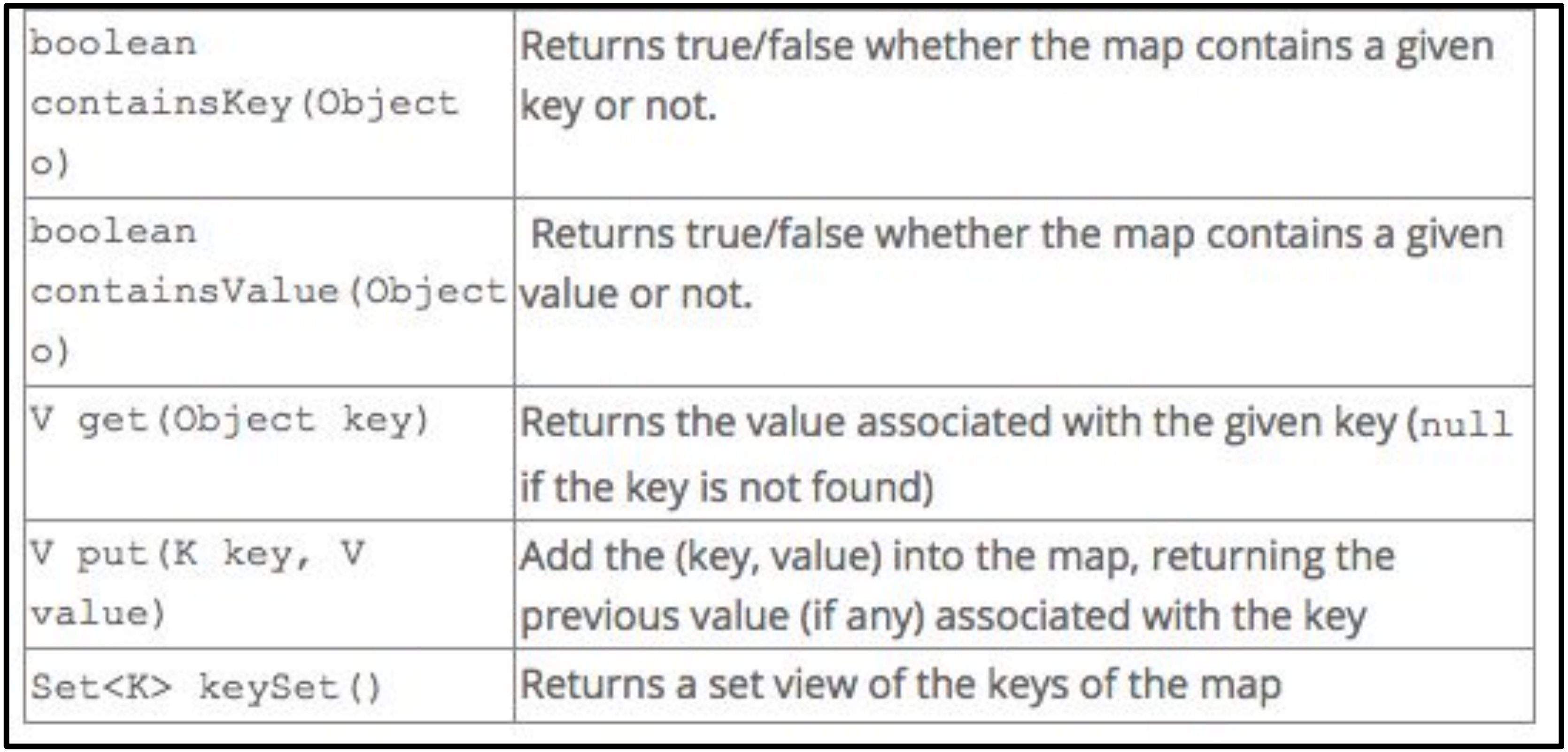
- Implementations:
  - `HashSet`
  - `LinkedHashSet`
  - `TreeSet`

The Map Interface

- Enables the definition of a mapping of keys to values

The Map Interface

METHODS

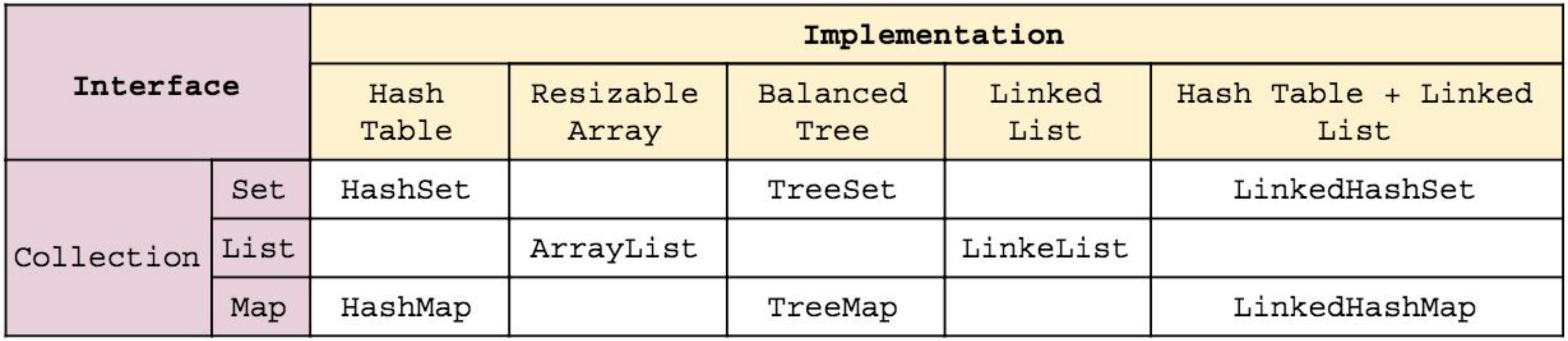


boolean containsKey(Object o)	Returns true/false whether the map contains a given key or not.
boolean containsValue(Object o)	Returns true/false whether the map contains a given value or not.
V get(Object key)	Returns the value associated with the given key (null if the key is not found)
V put(K key, V value)	Add the (key, value) into the map, returning the previous value (if any) associated with the key
Set<K> keySet()	Returns a set view of the keys of the map

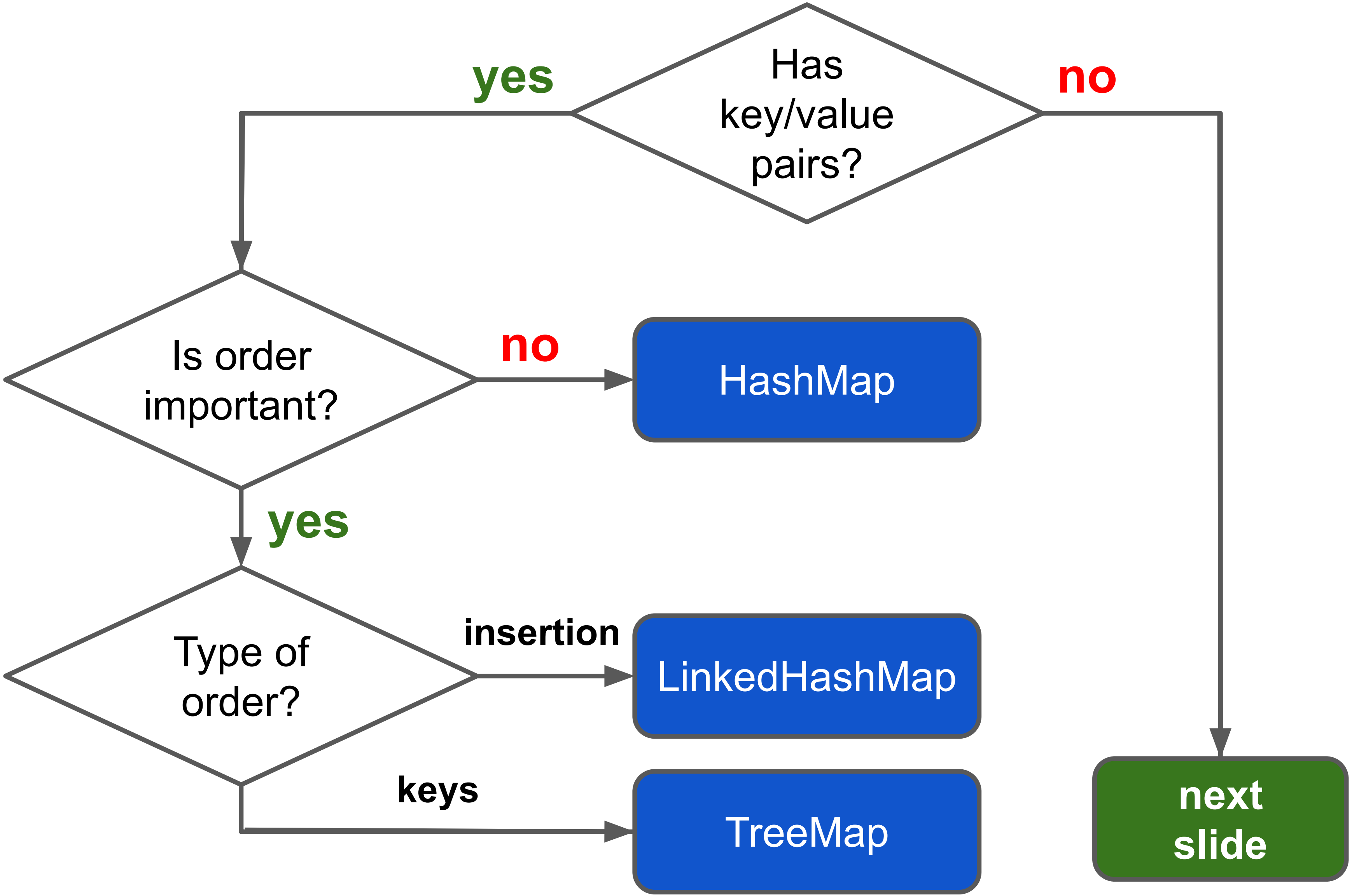
The Map Interface

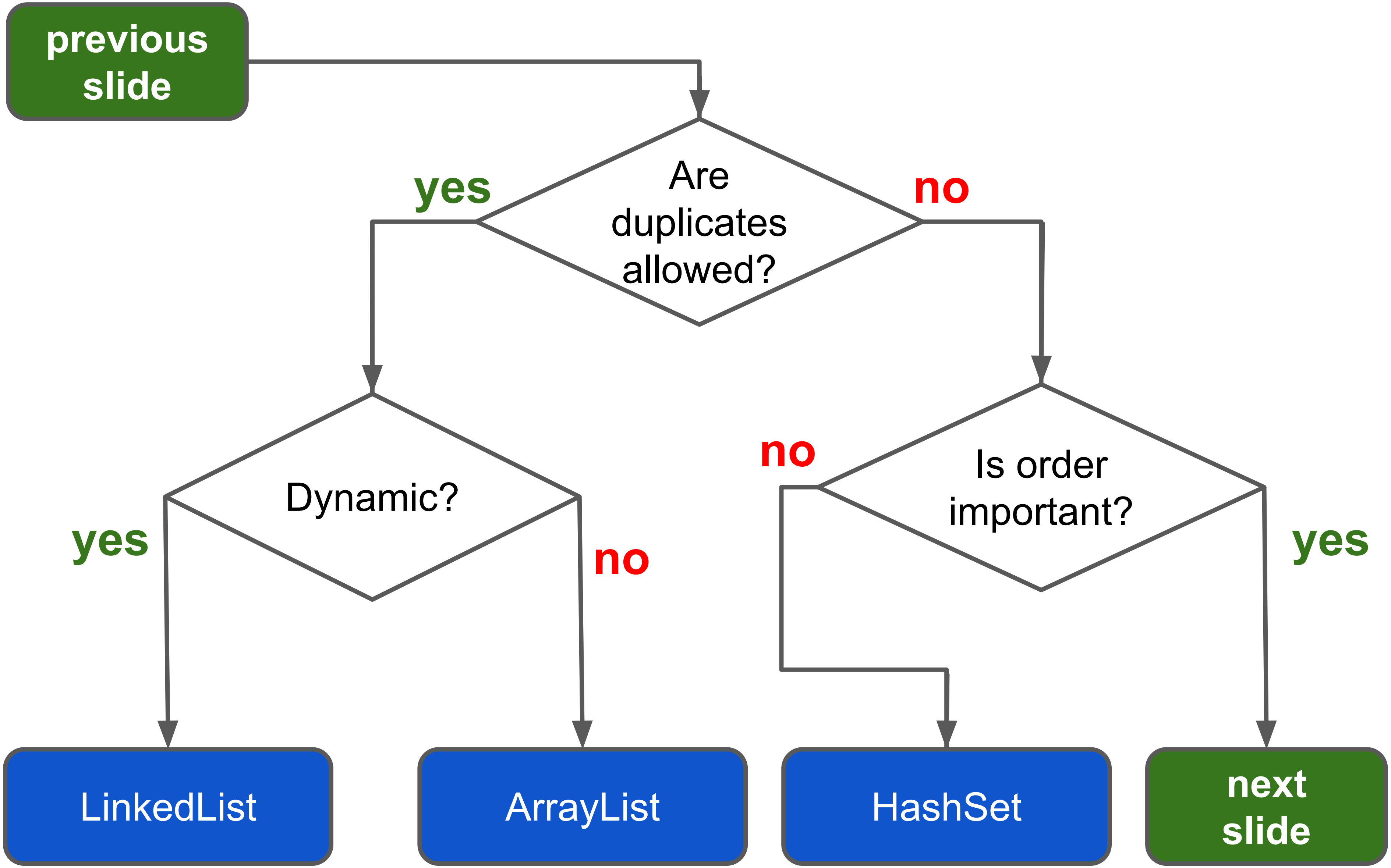
- Implementations:
  - `HashMap`
  - `LinkedHashMap`
  - `TreeMap`

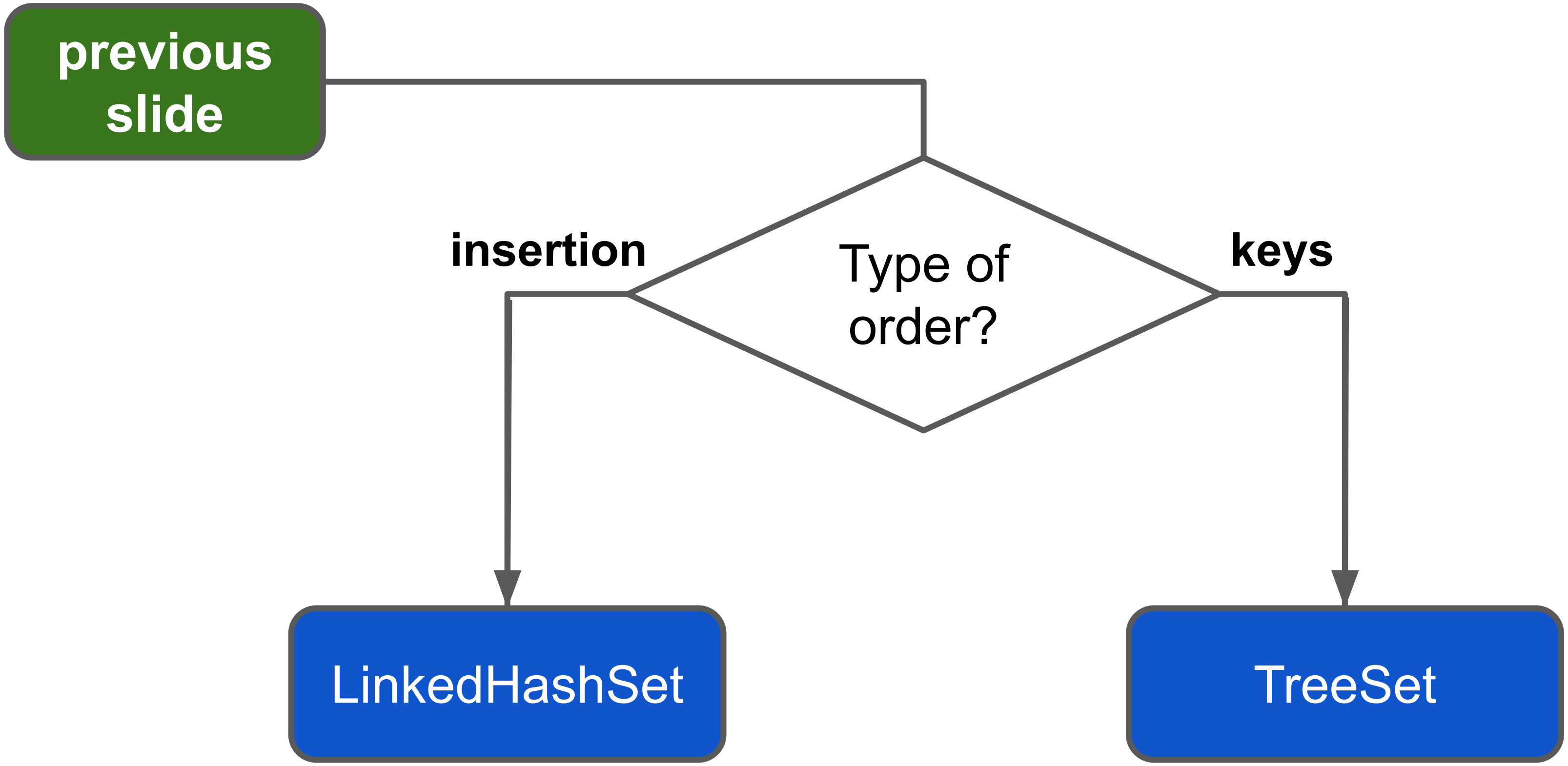
Summary



Interface		Implementation				
		Hash Table	Resizable Array	Balanced Tree	Linked List	Hash Table + Linked List
Collection	Set	HashSet		TreeSet		LinkedHashSet
	List		ArrayList		LinkeList	
	Map	HashMap		TreeMap		LinkedHashMap


Has key/value pairs?
yes
no
Is order important?
no
HashMap
yes
Type of order?
insertion
LinkedHashMap
keys
TreeMap
next slide


previous slide
Are duplicates allowed?
yes
no
Dynamic?
yes
no
Is order important?
no
yes
LinkedList
ArrayList
HashSet
next slide


previous slide

Type of order?

insertion

keys

LinkedHashSet

TreeSet

Practice

- To illustrate how the most important classes in the `Collections Framework` differ from each other, finish the TO-DO's in the activity called "collections" available on GitHub