Binghamton
University

CS-140
Fall 2020

Collections

Collection Basics

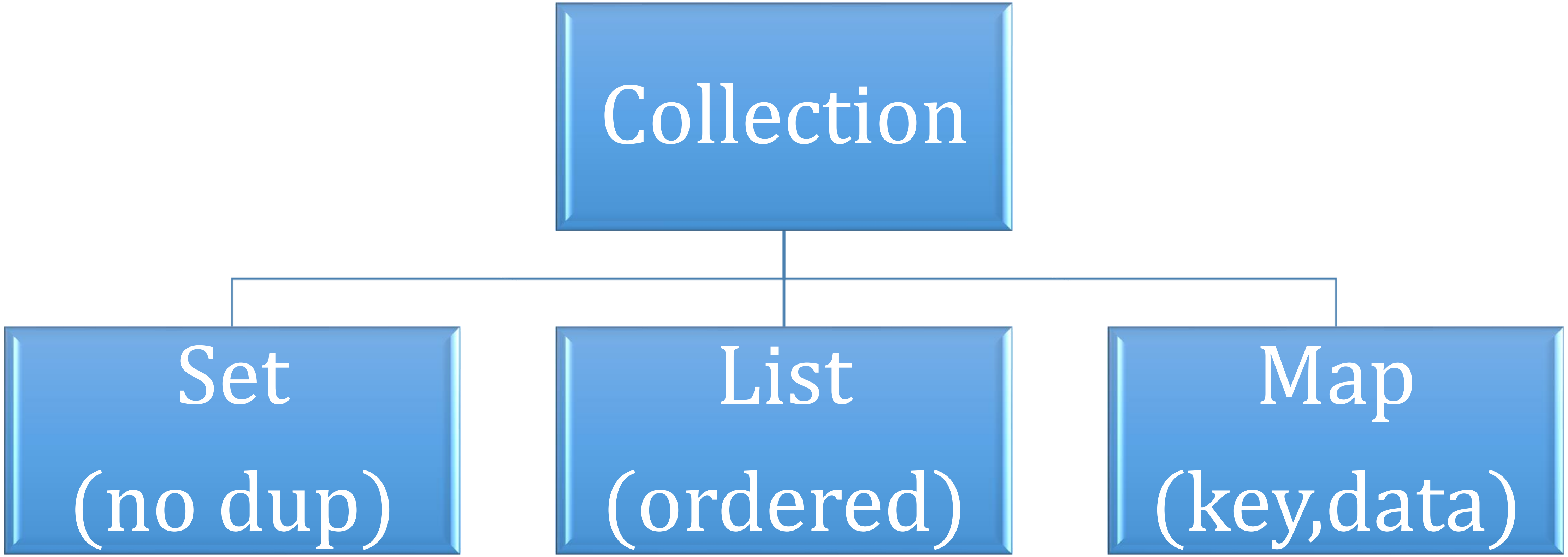
- What is a Collection?

  A group of objects
- What is the Java Collection Infrastructure?
  - Interfaces and Abstract Classes
  - Algorithms (provided as static methods) to manipulate collections
  - Some actual implementations
- Why have a Collection Infrastructure?
  - Provide pre-coded solutions for the most often used data structures
  - Standardize data structure access and usage
  - Provide high performance utilities to act on those structures
  - Enable Quick development of custom data structures

Collection Basics

- All Collections are generic
    - The elements of the collection can be any super-class of Object
    - All primitive types are auto-boxed into Integer, Double, etc.
- All collection implementation support a null constructor to make an empty collection, or a collection argument constructor to copy that collection.

The Java Collections Infrastructure



Collection

Set
(no dup)

List
(ordered)

Map
(key,data)

What kinds of methods?

- Creation
- Size/Add/Remove elements
- Comparison
- Traversal
- Extraction
- Conversion

The "Collection" interface

- Size/Add/Remove elements
  - add(e) addAll(c)
  - clear() remove(e) removeAll(c) removeIf(pred) reatainAll(c)
  - isEmpty() size()
- Traversal
  - iterator() spliterator() enhanced for loops
- Comparison
  - contains(e) containsAll(c) equals(o)
- Conversion
  - stream() parallelStream() toArray()

Collections static methods for Collections

- Comparison
  - disjoint(c1,c2)
- Extraction
  - frequency(c,o)
  - max(c) max(c,comp) min(c) min(c,comp)
- Conversion
  - synchronizedCollection(c) unmodifiableCollection(c)

Set interface

- Same as Collection interface, but sets have no duplicates
- Different implementations of add, addAll, and equals

Collections static methods for Sets

- Constructors
  - emptySet() singleton(e) newSetFromMap(m)
- Conversions
  - checkedSet(s) synchronizedSet(s) unmodifiableSet(s)

List Interface – Collection +

- Size/Add/Remove elements
  - add(index,e) addAll(index,c) set(index,e)
  - get(index)
  - remove(index)
  - replaceAll(operator)
- Traversal
  - listIterator() listIterator(index)
- Extraction
  - indexOf(e) lastIndexOf(e)
  - subList(from,to)
- Conversion
  - sort(comp)

Collections static methods for Lists

- Creation
  - emptyList() singletonList(e) list(enum)
  - copy(l1,l2) nCopies(n,e)
- Traversal
  - emptyListIterator()
  - fill(l,e) replaceAll(l,old,new) swap(l,i,j)
- Extraction
  - binarySearch(l,key,comp)
  - indexOfSublist(l,sl) lastIndexOfSublist(l,sl)
- Conversion
  - reverse(l) rotate(l,distance) shuffle(l) sort(l,comp)
  - checkedList(l) synchronizedList(l)

Map Interface

- A map consists of a set of keys linked to a list of values
  - Two generic types <K,V>
  - Each key is linked to a single value
- A map can be viewed as:
  - a set of keys, a collection of values, or a set of key/value pairs

Map Interface methods

- Creation
  - copyOf(m) of() of(k,v) of(k,v,k,v,…) ofEntries(…entries)
- Size/Add/Remove elements
  - put(k,v) putIfAbsent(k,v) putAll(m) get(k) getOrDefault(k,v)
  - merge(k,v,func)
  - remove(k) remove(k,v) clear()
  - isEmpty() size()
- Comparison
  - equals(o)
- Traversal
  - containsKey(k) containsValue(v)
  - forEach(consumer)
  - replace(k,v) replaceAll(func)
- Extraction
  - compute(k,f) computeIfAbset(k,f) computeIfPresent(k,f)
  - keyset() values()
- Conversion
  - entrySet()

Collections static methods for Maps

- Creation
  - emptyMap() singletonMap(k,v)
- Conversion
  - checkedMap(m,kt,vt) synchronizedMap(m) unModifiableMap(m)