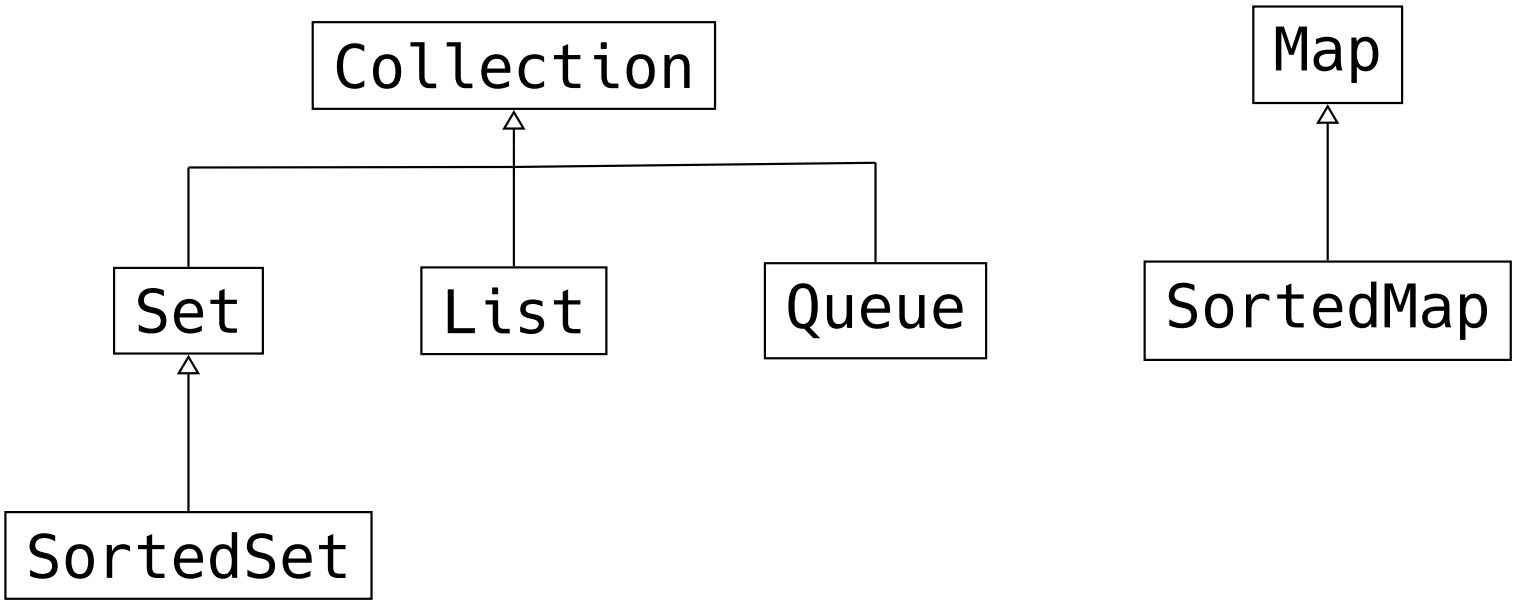
Java Collections Framework (JCF)

Collection

- A Collection is a group of things
  - These things/objects are called the elements of the collection
- A collection may or may not allow duplicates
  - Depends on the type of collection
- The *ordering* of a collection may or may not matter
- The different characteristics of a collection will define how it will perform (speed and space) for a given application
- A Collections Framework is a set of collections
  - Common interfaces that can be used somewhat interchangeably

The JCF



`Collection` declares useful methods for handling groups of objects

Collection Interface

- `Collection` provides:
  - `isEmpty`
  - `add`
  - `remove`
  - `size`
- `Set`
  - No duplicate items
  - Unordered
  - Implemented by `HashSet`, `TreeSet`
- `List`
  - Allows duplicate items
  - Ordered
  - Implemented by `ArrayList`, `LinkedList`, `Vector`
- `Queue`
  - Elements removed in the order they are inserted
  - Implemented by `AbstractQueue`

Map Interface

- Instead of only holding values, a map associates keys to values, allowing values to be retrieved by the key instead of an index
  - Implemented by: `HashMap`, `TreeMap`

AbstractCollection

- `AbstractCollection` implements Collection and provides the `toString()` method
  - Returns a string containing a representation of the elements in an order returned by the iterator
  - ex: `[01, 02, ...]`
- Iterators are returned by `.iterable()`, and provides:
  - `hasNext()`: returns true if there are more elements in the collection
  - `next()`: moves to the next element in the collection
- Iterators are what make the enhanced for loop possible

Abstract Data Types

- An Abstract Data Type (ADT) is a description of a way to contain and operate on data
  - The JCF implements these ADTs
- An ADT specifies:
  - How data is stored (Layout in memory)
  - How data is accessed (Ordered? Unordered?)
- This specification affects the performance (or possibility) of certain operations on the collection of data
  - A good choice of Data Structure can immensely speed up or slow down a program
- This semester, you will learn how different ADTs behave, implement some of the functionality in the ADTs, and see how to use the implementation provided by the JCF