Collections

- Common collection activities include adding objects, removing objects, verifying object inclusion, retrieving objects, and iterating.
- Three meanings for “collection”:
    - collection—Represents the data structure in which objects are stored
    - Collection—java.util.Collection—Interface from which Set and List extend
    - Collections—A class that holds static collection utility methods
- Three basic *flavors* of collections include *Lists, Sets, Maps*:
    - Lists of things: Ordered, *duplicates allowed,* with an *index*
    - Sets of things: May or may not be ordered and/or sorted, *duplicates not allowed*
    - Maps of things with keys: May or may not be ordered and/or sorted, *duplicate keys not allowed*
- Four basic *sub flavors* of collections include *Sorted, Unsorted, Ordered, Unordered*.
- Ordered means iterating through a collection in a specific, nonrandom order.
- Sorted means iterating through a collection in a *natural* sorted order.
- Natural means alphabetic, numeric, or programmer-defined, whichever applies.
- Key attributes of common collection classes:
    - ArrayList: Fast iteration and fast random access
    - Vector: Like a somewhat slower ArrayList, mainly due to its synchronized methods
    - LinkedList: Good for adding elements to the ends, i.e., stacks and queues
    - HashSet: Assures no duplicates, provides no ordering
    - LinkedHashSet: No duplicates; iterates by insertion order or last accessed
    - TreeSet: No duplicates; iterates in *natural* sorted order
    - HashMap: Fastest updates (key/value pairs); allows one null key,many null values
    - Hashtable: Like a slower HashMap (as with Vector, due to its synchronized methods). No null values or null keys allowed

- LinkedHashMap: Faster iterations; iterates by insertion order or last accessed, allows one null key, many null values
- TreeMap: A sorted map, in *natural* order