* List Interface
  + **ArrayList** : 상대적으로 빠르고 요소에 대해 순차적으로 접근할 수 있다.
  + **Vector** : ArrayList의 이전 버전이며 모든 메서드가 동기화 되어 있다.
  + **LinkedList** : 순서가 변경되는 경우 노드 링크만 변경하면 되므로 삽입, 삭제가 빈번 할 때 빠르다.
* Set Interface
  + **HashSet** : 빠른 접근 속도를 가지고 있으나 순서를 예측할 수 없다.
  + **LinkedHashSet** : 요소가 추가된 순서대로 접근할 수 있다.
  + **TreeSet** : 요소들의 정렬 방법을 직접 지정할 수 있다.
* Map Interface
  + **HashMap** : 중복을 허용하지 않고 순서를 보장하지 않으며 null 값을 허용한다.
  + **Hashtable** : HashMap 보다는 느리지만 동기화를 지원하며 null 값을 허용하지 않는다.
  + **TreeMap** : 정렬된 순서대로 Key와 Value를 저장하므로 빠른 검색이 가능하지만 요소를 추가할 때 정렬로 인해 오래걸린다.
  + **LinkedHashMap** : HashMap과 기본적으로 동일하지만 입력한 순서대로 접근이 가능하다.