Iterable  
Superclass semua collection di java yang bukan turunan dari map  
iterable digunakan untuk foreach collection, jadi semua collection yang turunan iterable pasti bisa di foreach  
  
Collection Interface  
Digunakan untuk memanipulasi data dari collection nya seperti tambah, hapus, edit  
Collection dibagi lagi menjadi list, set, dan queue  
Contoh method collection :  
- size  
- isEmpty  
- contains  
- toArray  
- clear  
- remove  
  
List Interface  
Struktur data collection yang punya sifat :  
- element bisa duplikat  
- data berurutan dengan kapan kita masukin  
- list pake index default number yang mirip kayak array  
List memiliki beberapa method kayak :  
- add  
- sort  
- remove  
- get  
- indexOf  
- set  
List dibagi menjadi 2:  
- ArrayList, implementasi dari list menggunakan array  
- LinkedList, implementasi dari linkedlist class dengan data yang disimpan dalam node dengan tersedianya next dan previous. gada info array nya  
List bisa diubah sifatnya menjadi mutable maupun immutable  
Mutable : datanya tidak bisa diubah lagi isi elemennya  
Immutable : datanya masih bisa diubah-ubah  
Ada beberapa beberapa method untuk membuat immutable list  
- Collections.emptyList() = list kosong  
- Collections.singletonList(e) = immutable 1 element  
- Collections.unmodifiableList(list) = konversi mutable ke immutable  
- List.of(e..) = immutabli dari element-element  
  
Set Interface  
Set merupakan collections yang berisikan element uniq yang berarti tidak duplikat  
Set tidak punya index kayak list, jadi gabakal urut  
Karena ga punya index, untuk ambil datanya harus di iterate satu persatu  
Hashtable, menyimpan hashcode  
Set memiliki 3 turunan:  
- EnumSet, tapi set ini jarang digunakan  
- HashSet, data tidak terurut dengan waktu kapan memasukan data. Kalo ga butuh terurut bisa pake ini  
- LinkedHashSet, data terurut dengan waktu kapan memasukan data. Kalo butuh urut pake ini  
Dalam set juga bisa diubah menjadi immutable kayak list, bedanya kalo elements pakenya Set.of bukan List.of  
  
SortedSet Interface  
Ini kayak set biasa tapi pas dimasukkin nanti otomatis secara otomatis  
Kalo elementnya bukan turunan dari comparable maka harus dibuat comparator untuk melakukan sorting nya  
  
NavigableSet Interface  
Turunan dari SortedSet  
Menambahkan method untuk beberapa operasi seperti :  
- lower  
- floor  
- ceiling  
- higher  
- poolFirst  
- poolLast  
Bisa immutable :  
- Collections.emptyNavigableSet()  
- Collections.unmodifiableNavigableSet(set)  
  
Queue Interface  
Collections antrian FIFO (First In First Out)  
Ada beberapa method tambahan di queue :  
- add, kalo udah mencapai kapasitas bakal throw error  
- offer, kalo udah sampe kapasitasnya cuma bakal return false  
- remove, mengambil dan menghapus data yang pertama kali dimasukkan, kalo gada data lagi bakal throw error  
- poll, sama kayak remove tapi return null  
- element, ambil data pertama tapi tidak dihapus, kalo data kosong bakal throw exception  
- peek, kayak element tapi return null  
Queue mempunyai turunan :  
- Dequeue  
- ArrayDequeue, menggunakan array sebagai implementasi queue nya, kayak arrayList  
- PriorityQueue, menggunakan array tapi diurutkan menggunakan comparabel atau comparator  
- LinkedList, menggunakan List dan gabakal ada batasnya  
  
Dequeue Interface  
Bisa dari depan maupun belakang, implementasi FIFO dan LIFO (Last In First Out)  
Seperti antrian dan tumpukan  
Deque bisa menggunakan LinkedList dan ArrayDequeue  
  
Map Interface  
Collections yang berisikan mapping key dan value  
1 key cuma boleh ke 1 value  
Kayak array tapi index nya bebas ga cuma angka  
Beberapa method dari map :  
- size  
- isEmpty  
- containsKey  
- containsValue  
- get, mengambil data berdasarkan key  
- put, nyimpen data  
- remove  
- keySet, ambil semua key  
- values, ambil semua value  
- entrySet, ambil key dan value  
Ada 5 implementasi Map :  
- HashMap  
impelmentasi dari map dengan menggunakan hashCode function dengan mengecek kesamaan menggunakan equals  
- WeakHashMap  
sama kayak HashMap tapi menggunakan weak key dimana jika tidak digunakan lagi maka datanya akan dihapus key nya.  
Cocok diganakan pada cache  
- IdentityHashMap  
Kayak HashMap tapi berbeda cara mengecek kesamaan datanya menggunakan method ==  
- LinkedHashMap  
Menggunakan LinkedList, datanya juga berurutan sesuai dengan kapan dimasukkan  
Proses get nya akan semakin lambat karena harus di loop satu persatu  
Kalo ambil data pake key lebih baik menggunakan HashMap biasa  
- EnumMap  
keynya menggunakan ENUM karena ENUM pasti uniq dan bakal lebih baik dari menggunakan Hash  
  
Immutable Map  
Map bisa dikonversi menjadi immutable seperti list dan set  
Method :  
- Collections.emptyMap()  
- Collections.unmodifiableMap(map)  
- Collections.singletonMap(key, value)  
- Map.of  
  
SortedMap Interface  
Impelmentasi Map dengan data key nya yang diurutkan menggunakan comparable maupun dibuat comparatornya  
SortedMap bisa mneggunakan semua method yang ada di Map  
Ada beberapa method tambahan seperti :  
- comparator(), memasukkan comparatornya  
- subMap(), memotong dari posisi key awal sampai terakhir  
- headMap(), ambil dari awal  
- tailMap(), ambil dari belakang  
Immutable SortedMap :  
- emptySortedMap  
- unmodifiableSortedMap  
  
NavigableMap  
Turunan dari SortedMap  
Bisa melakukan operasi kayak NavigableSet tapi isinya key dan value  
  
Enrtry  
Data dalam map disimpan dalam pair (key-value)  
Entry ini interface sederhana untuk mengambil key dan value  
  
LegacyCollections  
Terdiri dari :  
- Vector Class yang mirip sekali dengan ArrayList tapi methodnya menggunakan kata kunci synchronized yang thread safe  
- HashTable Class yang mirip dengan HashMap  
- Stack Class yang mirip dequeue yang LIFO  
  
Sorting  
Berbagai algoritma sorting sudah disediakan oleh java. sorting ini hanya bisa digunakan pada List karena pada collection lainnya sudah disediakan  
Method :  
- sort(list)  
- sort(list, comparator)  
  
BinarySerach  
Algoritma pencarian yang lebih cepat dari search bawaan java di list yang menggunakan sequential serach yang mencari satu persatu  
Penggunaan binary search ini harus menggunakan list yang sudah di sorting  
Menggunakan method :  
- Collectons.binarySearch(list, value)  
- Collectons.binarySearch(list, value, comparator)  
  
Collections Class  
Utility static method collection untuk manipulasi data collection  
Contoh :  
- copy(listTo, listFrom)  
- frequency(collection, object)  
- reverse(list)  
- suffle(list)  
- swap(list, from, to)  
  
Abstract Collection  
Semua algoritma dasar dari collection sudah ada abstract class nya karena algoritma tidak akan berubah  
Jika ingin membuat collection sendiri bisa implements abstract nya bukan interface nya  
Kumpulan abstract class :  
- AbstractCollection : Collection  
- AbstractList : List  
- AbstractMap : Map  
- AbstractQueue : Queue  
- AbstractSet : Set  
  
Default Method  
Berada pada interface baik dari List maupun Map, dengan penggunaan default method ini dapat mempersingkat penulisan kode jika sudah menggunakan lambda  
  
Spliterator Interface  
Interface untuk melakukan partisi data yang ada di collection dalam jumlah besar  
Nantinya akan di split dan diproses secara paralel nantinya pada multithread  
  
Konversi Array  
Untuk melakukan konversi ke array menggunakan toArray()  
Method :  
- Object[] toArray()  
- T[] toArray(new T[])