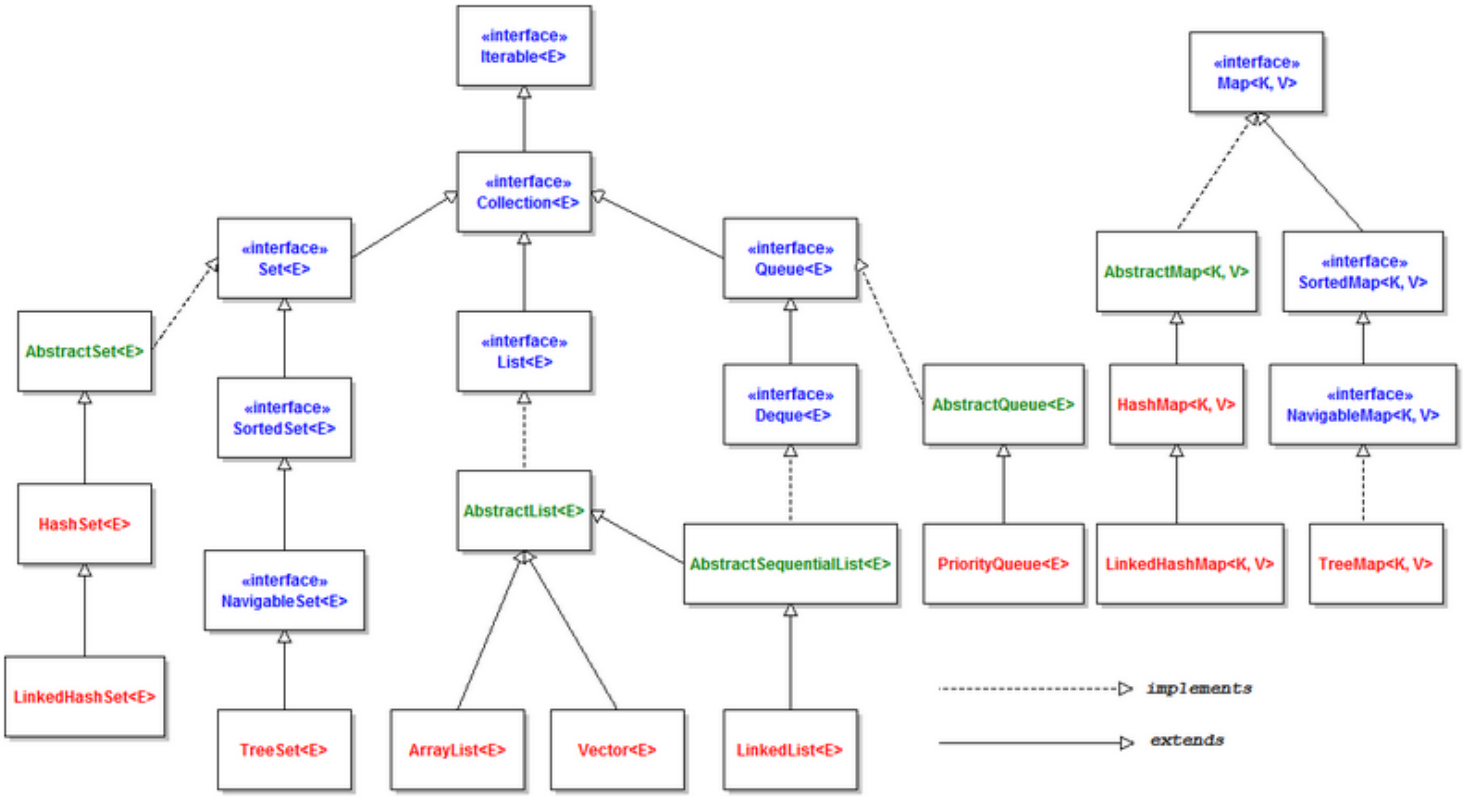
Programmation objet

Collections

Classes du framework Java



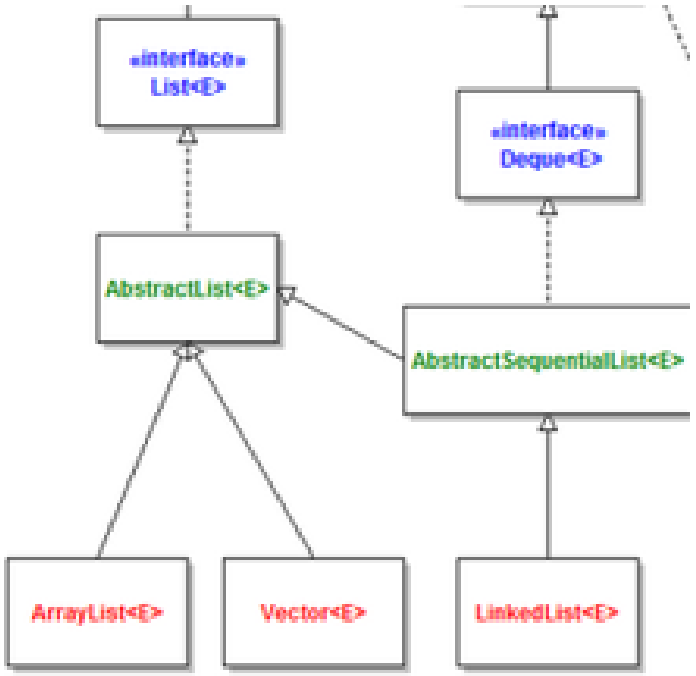
Class diagram of Java Collections framework

Intérêt de leur utilisation

- Facilité d'utilisation (même structure de classe)
- Méthodes validées
- Gestion automatique des manipulations sur les listes

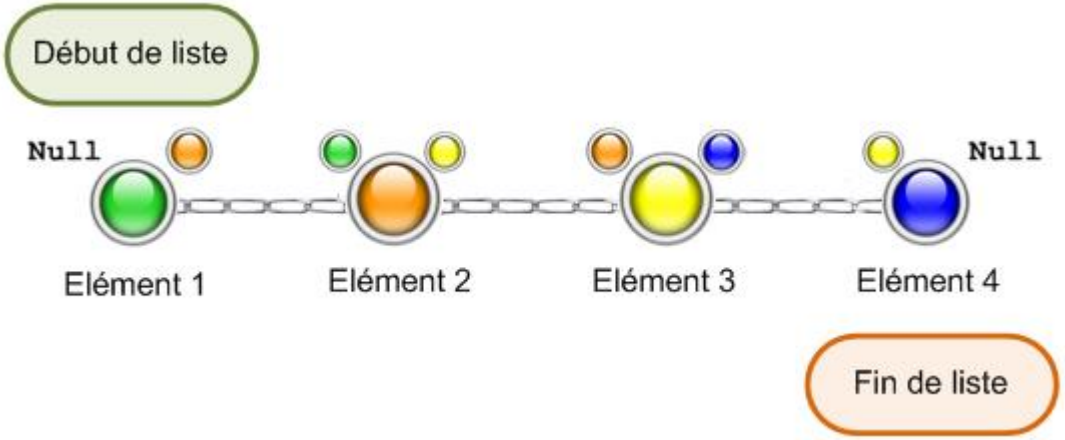
Collection type LIST

- Objets organisés sous le format d'un tableau
  - Indice pour accéder à un objet
  - Taille extensible



LinkedList

- Chaque élément contient une référence vers l'élément suivant et le précédent (valeur null si aucun)



ArrayList

Tableau classique

Ajout de tous types de valeurs

```
ArrayList tab = new ArrayList();
tab.add("test");
tab.add(3);
System.out.println(tab.get(0));
```

Tableau typé

Ajout d'objets Heros (ou sous-type) uniquement

```
ArrayList<Heros> tabHeros = new ArrayList<Heros>();
tabHeros.add(ht1);
tabHeros.add(hf1);
tabHeros.add(ht2);
tabHeros.add(hf2);

//test du polymorphisme avec toString()
for (Heros h : tabHeros)
    System.out.println(h.toString());
```

Comparatif

ArrayList

- Objets non liés les uns aux autres
- Plus efficace en lecture

LinkedList

- Liaisons entre objet
- Plus efficace en cas d'insertion/suppression en milieu de liste

Les itérateurs

- Création d'un objet associé à une liste pour faciliter le parcours
- Initialisé à partir des méthodes des classes List ou Set

Fonctionnement

Class ListIterator

Modifier and Type	Method and Description
void	add(E e) Inserts the specified element into the list (optional operation).
boolean	hasNext() Returns true if this list iterator has more elements when traversing the list in the forward direction.
boolean	hasPrevious() Returns true if this list iterator has more elements when traversing the list in the reverse direction.
E	next() Returns the next element in the list and advances the cursor position.
int	nextIndex() Returns the index of the element that would be returned by a subsequent call to next().
E	previous() Returns the previous element in the list and moves the cursor position backwards.
int	previousIndex() Returns the index of the element that would be returned by a subsequent call to previous().
void	remove() Removes from the list the last element that was returned by next() or previous() (optional operation).
void	set(E e) Replaces the last element returned by next() or previous() with the specified element (optional operation).

Class Iterator

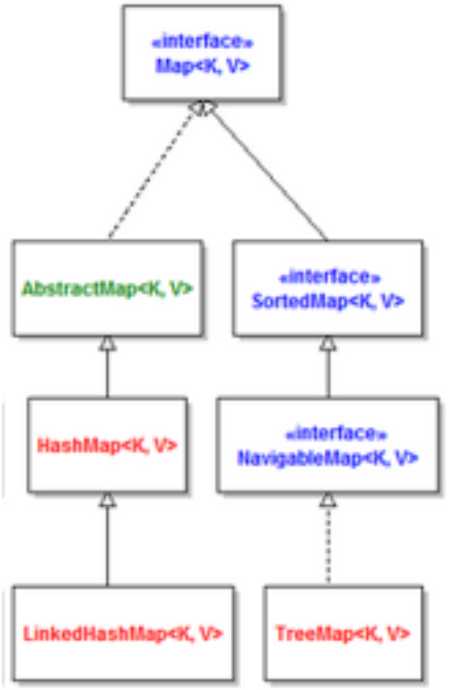
Modifier and Type	Method and Description
boolean	hasNext() Returns true if the iteration has more elements.
E	next() Returns the next element in the iteration.
void	remove() Removes from the underlying collection the last element returned by this iterator (optional operation).

Utilisation d’itérateur

- Créer un tableau de produits
- Ajouter 5 produits
- Utiliser un itérateur pour afficher la liste des produits

Collection type MAP

- Liste basée sur des couples clé-valeur
  - Permet de chercher une valeur à partir de sa clé



Création d'un objet HashMap

- Définit le type de la clé et le type de la valeur à la création de l'objet

```
HashMap <typeClé, typeValeur> laTable = new HashMap();
```

Méthodes de la classe

Modifier and Type	Method and Description
void	clear() Removes all of the mappings from this map.
Object	clone() Returns a shallow copy of this HashMap instance: the keys and values themselves are not cloned.
boolean	containsKey(Object key) Returns true if this map contains a mapping for the specified key.
boolean	containsValue(Object value) Returns true if this map maps one or more keys to the specified value.
Set<Map.Entry<K,V>>	entrySet() Returns a Set view of the mappings contained in this map.
V	get(Object key) Returns the value to which the specified key is mapped, or null if this map contains no mapping for the key.
boolean	isEmpty() Returns true if this map contains no key-value mappings.
Set<K>	keySet() Returns a Set view of the keys contained in this map.
V	put(K key, V value) Associates the specified value with the specified key in this map.
void	putAll(Map<? extends K,? extends V> m) Copies all of the mappings from the specified map to this map.
V	remove(Object key) Removes the mapping for the specified key from this map if present.
int	size() Returns the number of key-value mappings in this map.
Collection<V>	values() Returns a Collection view of the values contained in this map.

Gestion des produits

Création des objets

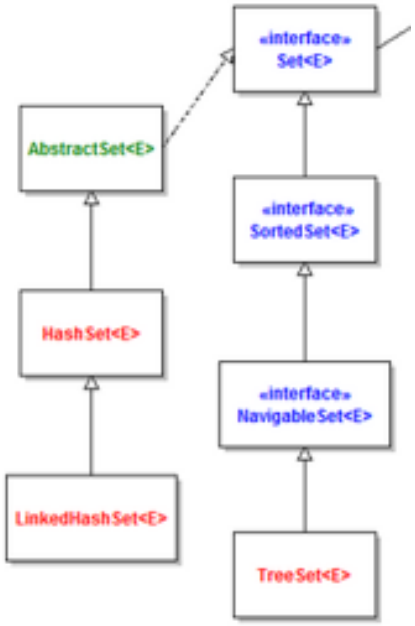
- Créer une collection de type HashMap qui contiendra la liste des produits associés avec comme clé la référence produit (type String)
- Ajouter des produits dans la liste

Appel des méthodes

- Parcourir la liste avec un itérateur
- Demander une référence produit à l'utilisateur, affiche la description du produit ou un message d'erreur
- Demander une référence produit à l'utilisateur, supprimer ce produit ou afficher un message d'erreur

Collection : type SET

- Liste d'objets sans doublons
  - Liste classique HashSet
  - Liste triée TreeSet



Utilisation de TreeSet

- Les objets stockés dans le TreeSet doivent être comparables
  - Pour une nouvelle classe, implémenter l'interface Comparable et définir la méthode compareTo(Object o)

Utilisation de TreeSet

- Créer une liste contenant les noms des étudiants
  - Ordre aléatoire
  - Insérer 2 fois le même nom
- Parcourir la liste à l'aide d'un itérateur pour afficher les noms dans l'ordre alphabétique
- Afficher les noms dans l'ordre inverse