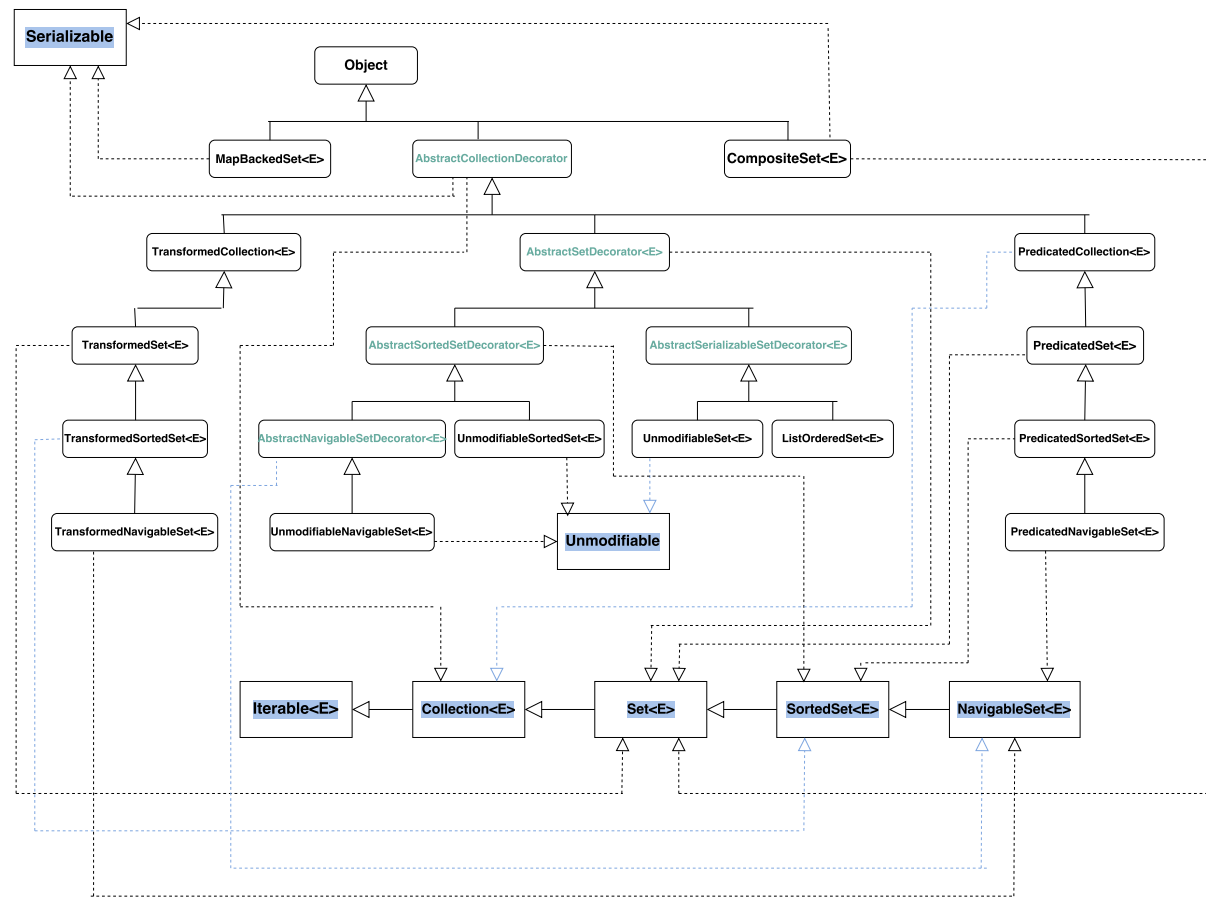


TP1 Extraction d'une hiérarchie d'interfaces Java

Tamara ROCACHER

Anastasiia PRYSIAZHNIUK

Compréhension d la hirarchie initiale



Extraction de données de base

cf extractionInterface2017.java, code commenté.

Extraction d'une hiérarchie d'interface et analyse

```

{[[Hiérarchie d'Interfaces], méthode ] = [les indices de classes où cette méthode est implémentée]]
{[[Unmodifiable, SortedSet, Set, Collection, Serializable], boolean add(java.lang.Object) ]=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]]
{[[Unmodifiable, SortedSet, Set, Collection, Serializable], boolean remove(java.lang.Object) ]=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]]
{[[Unmodifiable, SortedSet, Set, Collection, Serializable], void clear() ]=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]]
{[[Unmodifiable, SortedSet, Set, Collection, Serializable], boolean isEmpty() ]=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]]
{[[Unmodifiable, SortedSet, Set, Collection, Serializable], boolean contains(java.lang.Object) ]=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]]
{[[Unmodifiable, SortedSet, Set, Collection, Serializable], int size() ]=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]]
{[[Unmodifiable, SortedSet, Set, Collection, Serializable], class [Ljava.lang.Object; toArray() ]=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]]
{[[Unmodifiable, SortedSet, Set, Collection, Serializable], class [Ljava.lang.Object; toArray([Ljava.lang.Object;) ]=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]]
{[[Unmodifiable, SortedSet, Set, Collection, Serializable], interface java.util.Iterator iterator() ]=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]]
{[[Unmodifiable, SortedSet, Set, Collection, Serializable], boolean addAll(java.util.Collection) ]=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]]
{[[Unmodifiable, SortedSet, Set, Collection, Serializable], boolean containsAll(java.util.Collection) ]=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]]
{[[Unmodifiable, SortedSet, Set, Collection, Serializable], boolean removeAll(java.util.Collection) ]=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]]
{[[Unmodifiable, SortedSet, Set, Collection, Serializable], boolean retainAll(java.util.Collection) ]=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]]
{[[Unmodifiable, SortedSet, Set, Collection, Serializable], default interface java.util.Spliterator spliterator() ]=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]]
{[[Unmodifiable, SortedSet, Set, Collection, Serializable], default interface java.util.stream.Stream stream() ]=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]]
{[[Unmodifiable, SortedSet, Set, Collection, Serializable], default boolean removeIf(java.util.function.Predicate) ]=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]]
{[[Unmodifiable, SortedSet, Set, Collection, Serializable], default interface java.util.stream.Stream parallelStream() ]=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]]
{[[Unmodifiable, SortedSet, Set, Collection, Serializable], default void forEach(java.util.function.Consumer) ]=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]]
{[[Unmodifiable, NavigableSet, SortedSet, Set, Collection, Serializable], class java.lang.Object pollFirst() ]=[3, 6, 9]]
{[[Unmodifiable, NavigableSet, SortedSet, Set, Collection, Serializable], class java.lang.Object pollLast() ]=[3, 6, 9]]

```

A partir des résultats obtenus précédemment (extrait: image ci dessus), nous observons des groupes de méthodes (une ligne = une méthode) qui se retrouvent dans des groupes de classes concrètes (de 0 à 11: indices de nos classes concrètes). Nous avons regroupé ces méthodes en fonction des classes les implémentant, puis observé les classes communes et supplémentaires pour chaque groupe de méthodes afin de déterminer lesquelles sont héritées et à quel niveau. Par exemple, le premier groupe de méthodes (image ci dessus) se retrouve dans toutes les classes, soit directement (implémentation de l'interface) soit par héritage. Ce groupe correspond donc à l'interface Collection et se trouve tout en haut de notre hiérarchie. Nous obtenons la hiérarchie suivante:

