Examen ECRIT - session 1 - Documents sur le projet Collections autorisés

Questions sur le projet d'analyse des collections Java

Dans cette partie, nous vous proposons de travailler sur une analyse des résultats produits par les différents groupes parmi lesquels vous pourrez reconnaître le vôtre grâce aux initiales. Mettez-vous dans une position d'analyse constructive, il ne s'agit pas de critiquer des propositions mais de comprendre pourquoi elles sont faites et ce qu'elles peuvent apporter comme point de vue. Au passage, vous pouvez indiquez si les données de vos travaux sont bien telles que décrites ci-dessous.

Partie des collections analysée

Le schéma de la figure 1 vous présente les classes et les interfaces de la hiérarchie d'origine, sans les collections ni les maps du paquetage `java.util.concurrent` (que vous pourrez devinez à leur préfixe dans la suite). Il vous aidera à vous repérer. Les classes `SubList` et `RandomAccessSubList` sont des classes non publiques incluses dans le fichier `AbstractList.java`.

Question 1. Les choix qui ont été faits dans les différents groupes sont classés dans des treillis présentés dans les figures 2 et 3. **Travaillez uniquement sur le treillis de gauche** de la figure 2 qui vous montre les classes abstraites sélectionnées pour étude par les différents groupes : décrivez les grandes options que vous comprenez, commentez celle que vous avez faite par rapport aux autres.
Nota Les deux autres treillis (figure 2 droite et figure 3) vous sont donnés pour information et ne sont pas nécessaires pour répondre aux questions. Pour les curieux qui liront les autres treillis après l'examen : la classe `Properties` représente un ensemble persistant de propriétés, elle dérive de la classe `Hashtable`. La classe `Bitset` représente un vecteur de bits et ne dérive que des deux interfaces `Cloneable`, et `Serializable`. Les autres classes sont très spécifiques : "The Attributes class maps Manifest attribute names to associated string values" ; "ArraysParallelSortHelpers and ArrayPrefixHelpers contain Helper utilities for the parallel sort methods in Arrays.parallelSort" ; "A RoleUnresolvedList represents a list of RoleUnresolved objects, representing roles not retrieved from a relation due to a problem encountered when trying to access (read or write) the roles." Les classes `sort` implémentent des tris..

Données extraites et relations construites pour l'étude

La figure 4 présente les choix de données effectués : à la fois quelles données ont été extraites et quelles relations ont été construites à partir de ces données. Certains de vos documents `pdf` ne contenaient pas les mêmes informations que vos fichiers de données `rcft`, dans ce cas, les données des fichiers `rcft` ont été utilisées.

Pour lire le diagramme :

- "AE" représente les types d'entités analysées (lignes dans les tables)
- `Abs.Conc.X.finalStaticFields.methodsNames` représente la relation dont les lignes sont les classes abstraites et concrètes et dont les colonnes (après le 'X') sont les attributs static final et les noms des méthodes (à transposer pour les autres relations)
- "itf" veut dire interface
- "allImplement.itf" veut dire : toutes les interfaces implémentées sont en colonne
- "implemented.itfOutOfCollections" veut dire : seules les interfaces implémentées et qui sont hors du paquetage Collection sont en colonne (par exemple Cloneable, mais pas `List`)
- "excep" veut dire "exception"

Question 2.
a- Beaucoup de groupes ont utilisé les attributs static finaux privés. Comment cela se justifie-t-il lorsque l'on recherche des interfaces ?
b- Le diagramme de la figure 4 présente une grande variation dans les choix. Ne cherchez pas à tout commenter. Commentez 2 ou 3 grandes tendances que vous observez (par exemple quels sont les choix effectués le plus souvent, que vous voyez dans les concepts du haut) et situez vos choix parmi les autres en les réexpliquant.

Analyse des séquences

Question 3. Le diagramme de la figure 5 présente l'AOC-poset limité aux séquences (interfaces, classes abstraites et concrètes, non concurrentes), décrites par les signatures des méthodes publiques. Les méthodes publiques héritées de la classe `Object` ont été retirées pour se concentrer sur ce qui est propre aux collections.

- Comparez de manière globale la hiérarchie de l'AOC-poset avec la hiérarchie de classes et d'interfaces initiale (dans leur organisation et dans les relations de spécialisation).
- Analysez les concepts de l'AOC-poset. Indiquez lesquels vous trouvez intéressants pour analyser les classes et pourquoi (par exemple pour comprendre leur organisation). Chercher à nommer les concepts qui n'ont pas d'élément dans leur extension (partie basse de la boîte du concept vide, comme le concept `Concept_Collections_8`) d'après leur position ou leurs caractéristiques.
- Qu'observe-t-on sur la méthode `clone` ?
- Pourriez-vous tirer de l'AOC-poset de nouvelles interfaces ?

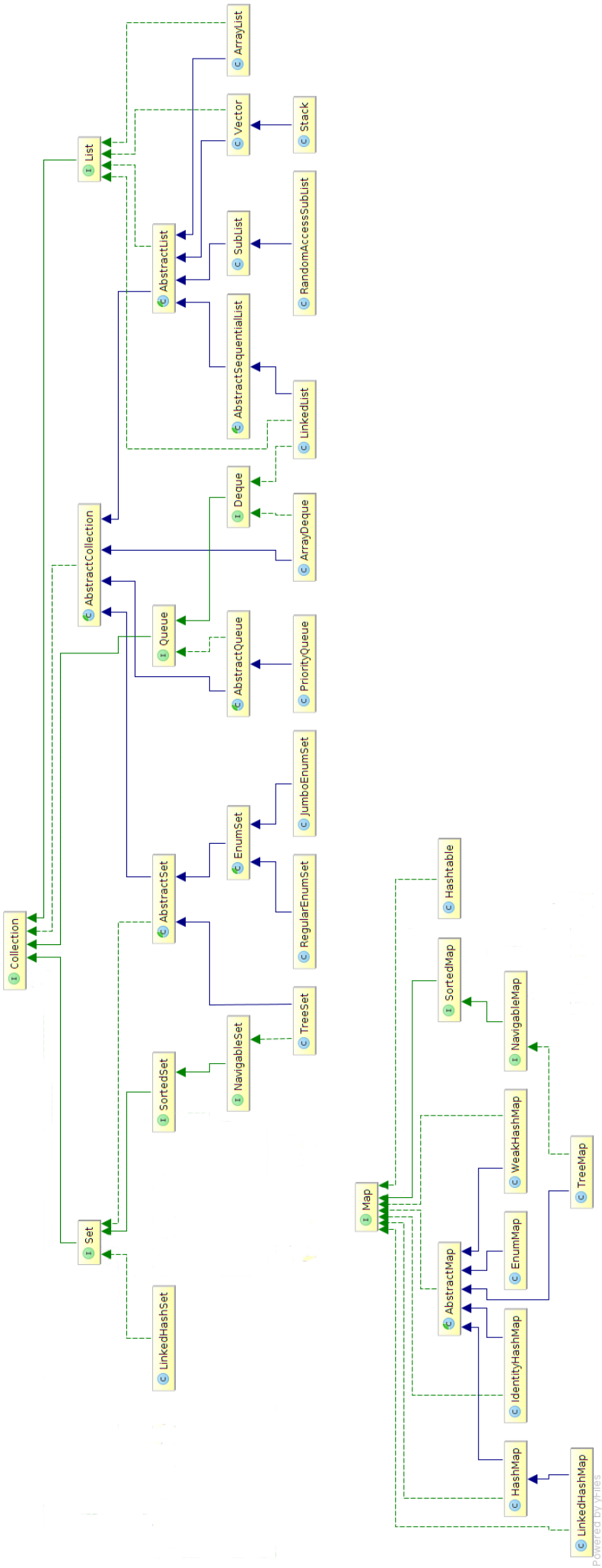

FIGURE 1 – Diagramme UML sans les structures concurrentes (par le groupe CVM)

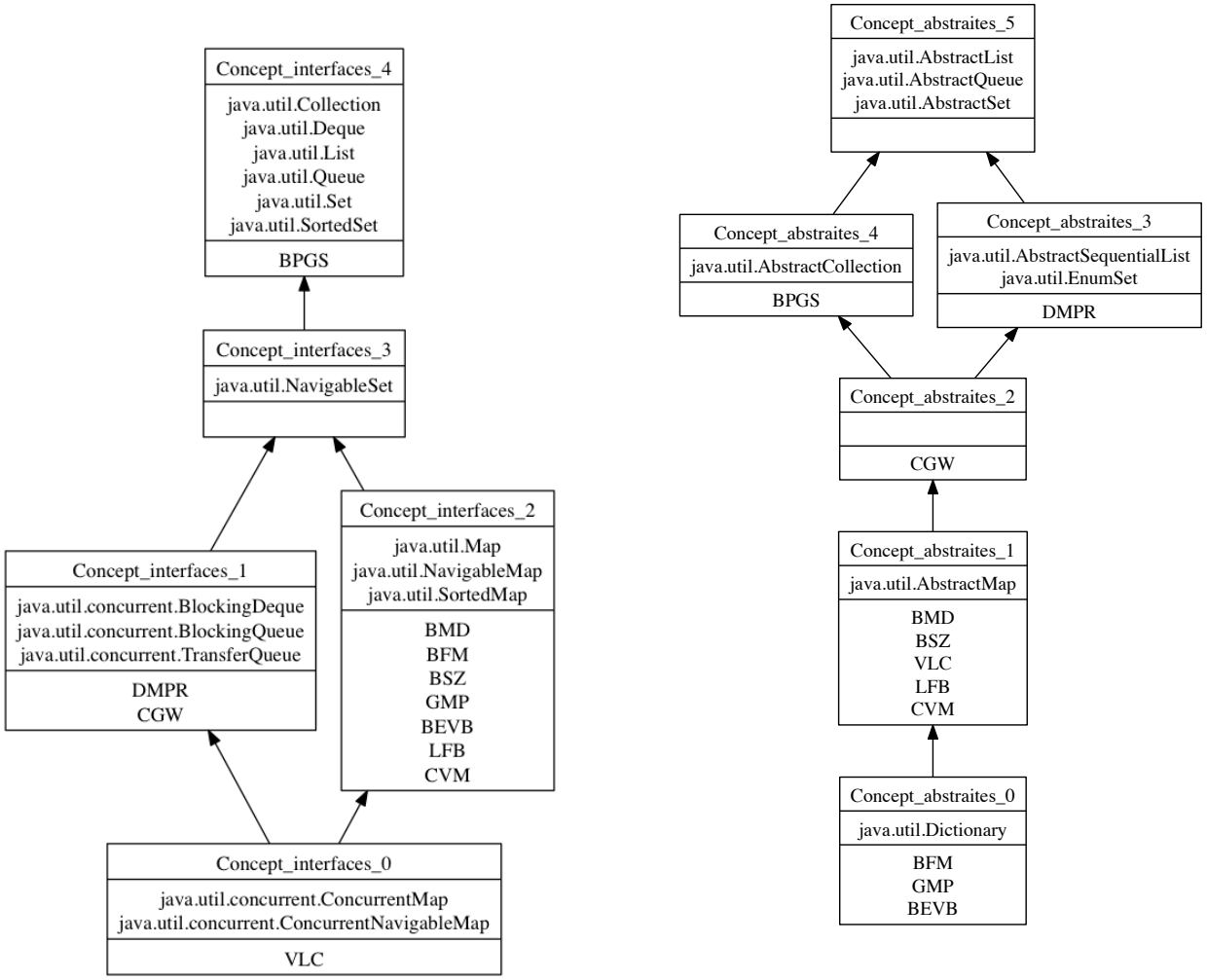

FIGURE 2 – à gauche : choix des interfaces par les différents groupes ; à droite : choix des classes abstraites

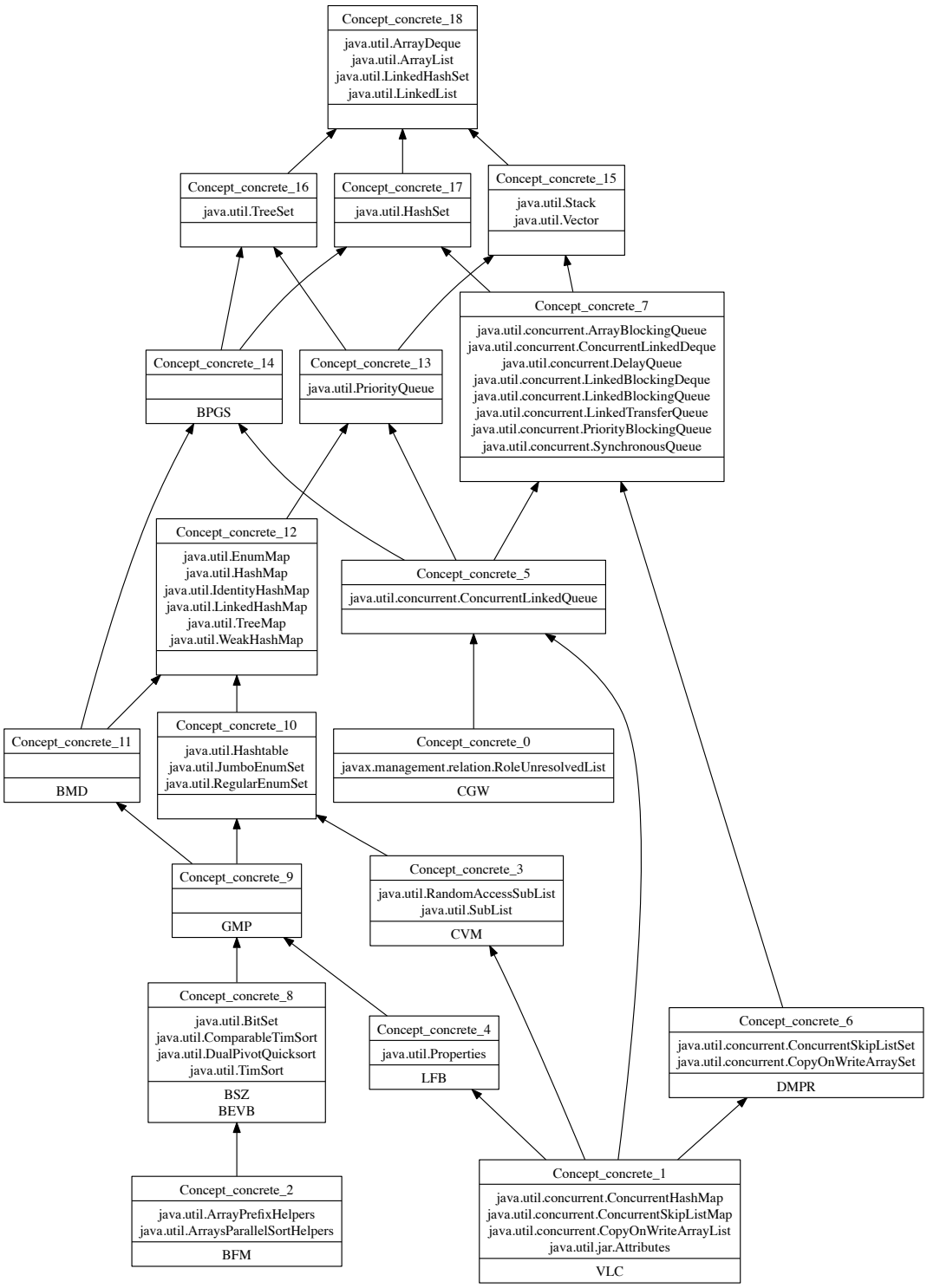

FIGURE 3 – Choix des classes concrètes par les différents groupes

FIGURE 4 – Choix des données extraites et des relations par les différents groupes

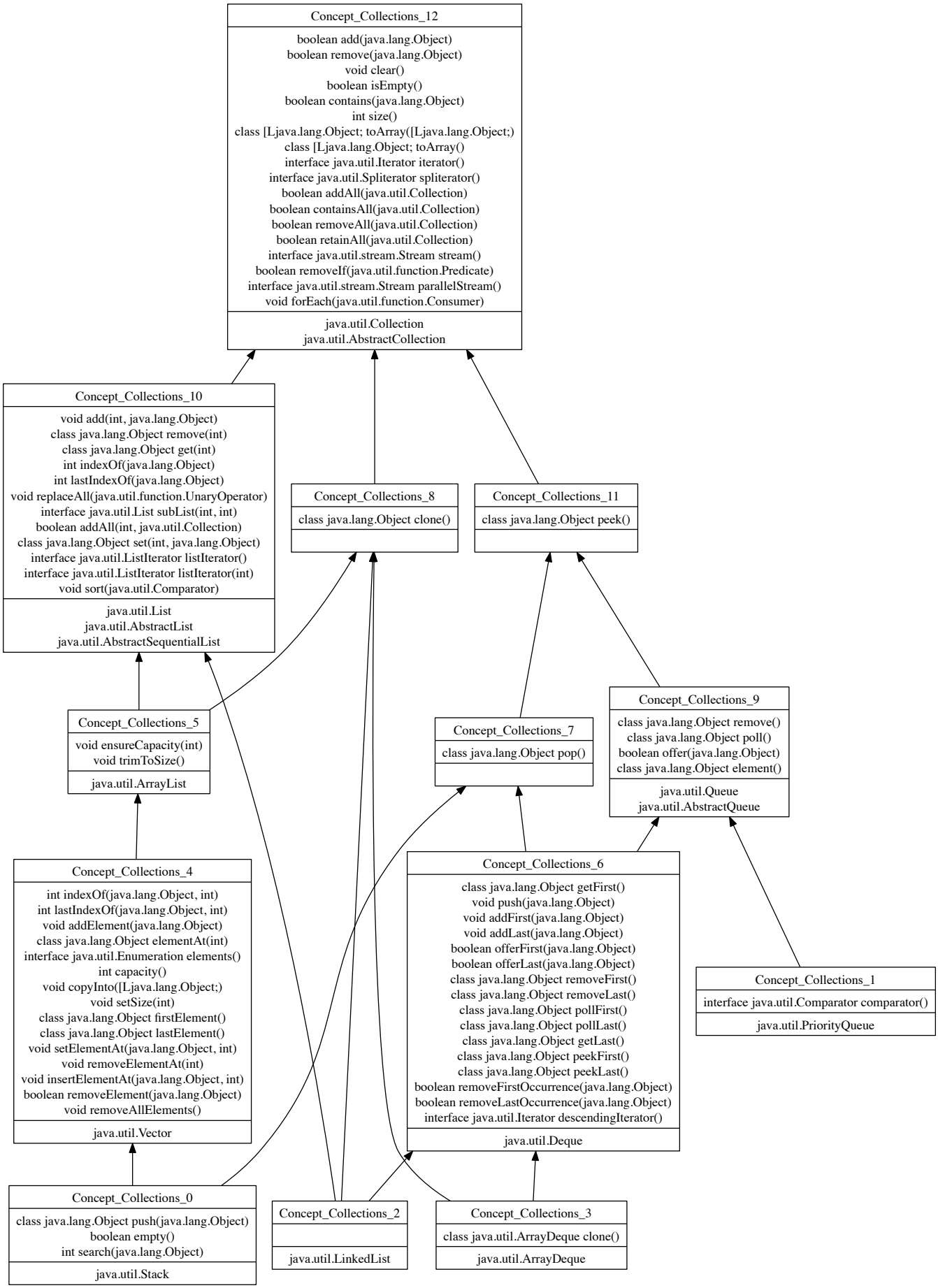

FIGURE 5 – Analyse des séquences