

Secteur Tertiaire Informatique
Filière « Etude et développement »

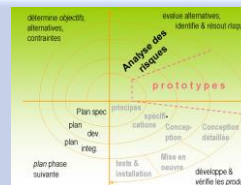
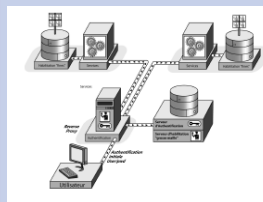
API Collection

POO

Apprentissage

Mise en situation

Evaluation



1. INTRODUCTION

L'objectif de cette activité est de prendre en main les structures de données communément utilisées en Java.

1.1 LES STRUCTURES DE DONNEES

Définition

Une structure de données est une organisation logique, construite et destinée à contenir des informations.

Cette structure, organisée donc en tant que conteneur, permet de globaliser, de simplifier et d'unifier les traitements réalisables sur les informations contenues.

Parmi les structures de données les plus usitées, que l'on appelle également des collections, on retrouve :

- Les tableaux,
- Les listes chaînées,
- Les piles, les files,
- Les graphes,
- Les arbres

A faire

Afin de commencer à étudier les structures de données vous pouvez lire le cours intitulé « cours-api-collection » disponible sur Métis (jusqu'à la page 13).

Dans ce cours des exemples de codes sont fournis mais il ne vous est pas nécessaire de les analyser en détails.

L'intérêt est de comprendre le fonctionnement général des structures de données suivantes :

- Les tableaux,
- Les listes chaînées,
- Les piles, les files,

1.2 API COLLECTION EN JAVA

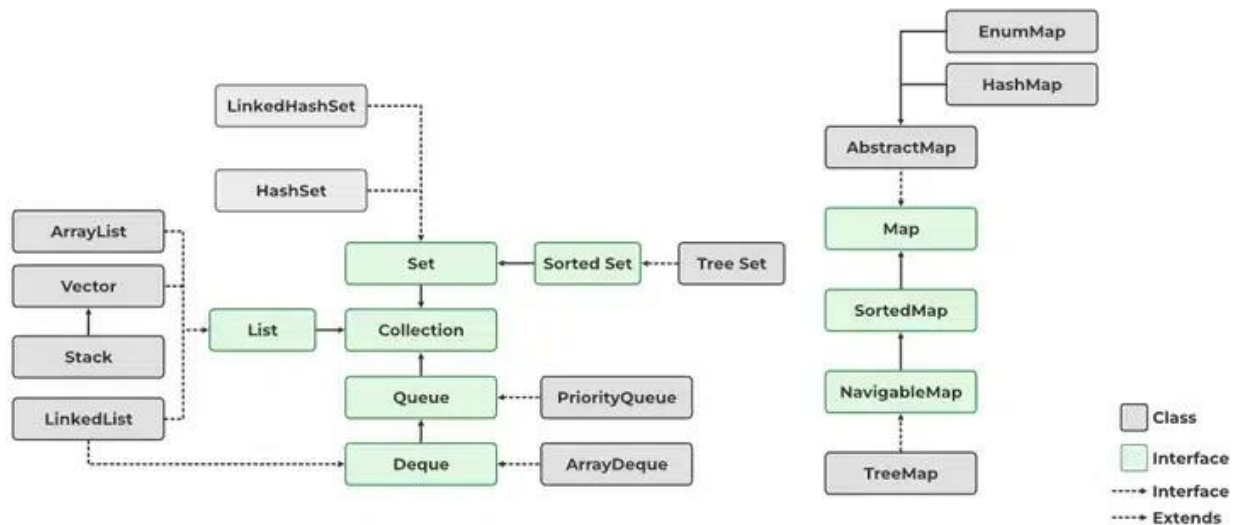


Figure 1 Hiérarchie des classes représentant l'API Collection

La JDK Java propose un ensemble de classes utilitaires qui implémentent les structures de données communes. Ces classes sont répertoriées sous l'appellation : API Collection.

Le diagramme de classes simplifié disponible ci-dessus présente les l'organisation des différentes classes de l'API Collection.

Les collections sont donc objets qui permettent de gérer des ensembles d'objets. Ces ensembles de données peuvent être définis avec plusieurs caractéristiques : la possibilité de gérer des doublons, de gérer un ordre de tri, etc. ...

Une collection est un regroupement d'objets qui sont désignés sous le nom d'éléments.

L'API Collections propose un ensemble d'interfaces et de classes dont le but est de stocker de multiples objets. Elle propose quatre grandes familles de collections, chacune définie par une interface de base :

- **List** : collection d'éléments ordonnés qui accepte les doublons
- **Set** : collection d'éléments non ordonnés par défaut qui n'accepte pas les doublons
- **Map** : collection sous la forme d'une association de paires clé/valeur
- **Queue et Deque** : collections qui stockent des éléments dans un certain ordre avant qu'ils ne soient extraits pour traitement.

L'activité proposée dans la suite de ce document vous permettra de faire vos premiers pas dans la compréhension des structures les plus utilisées.

2. ACTIVITE DE RECHERCHE

2.1 QUESTIONS

Afin de découvrir l'API Java Collection, répondez aux questions suivantes (bien entendu, des recherches vous seront nécessaires).

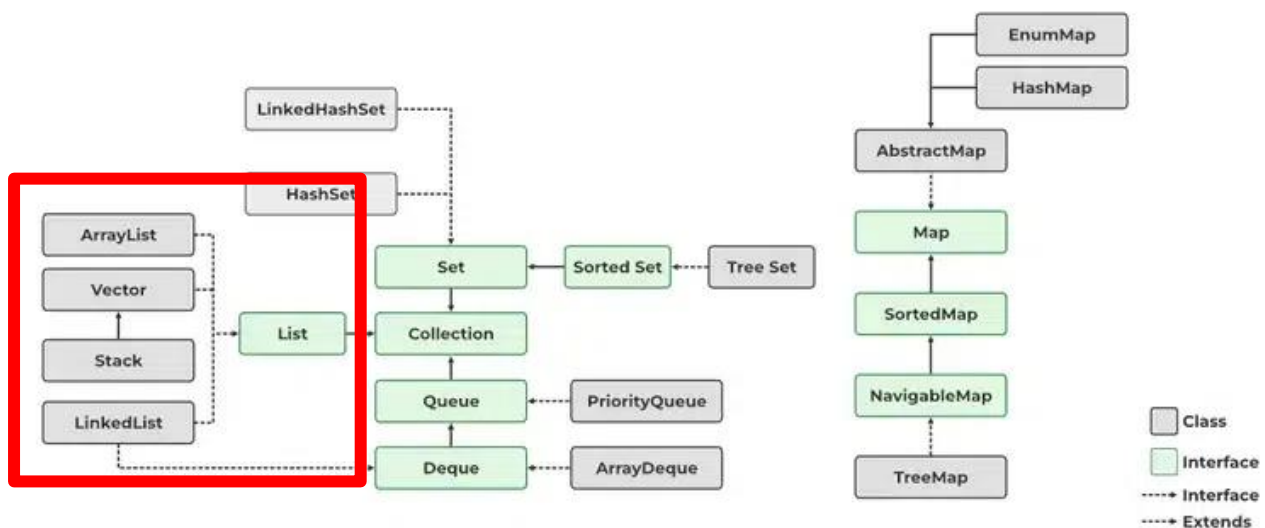
Attention

En plus de vos recherches, il vous faudra certainement analyser la documentation officielle du JDK Java (lien suivant pour la version 17) :

<https://docs.oracle.com/en/java/javase/17/docs/api/index.html>

2.1.1 Questions portant les classes qui implémentent l'interface « List »

Le diagramme suivant indique les classes que vous allez étudier :



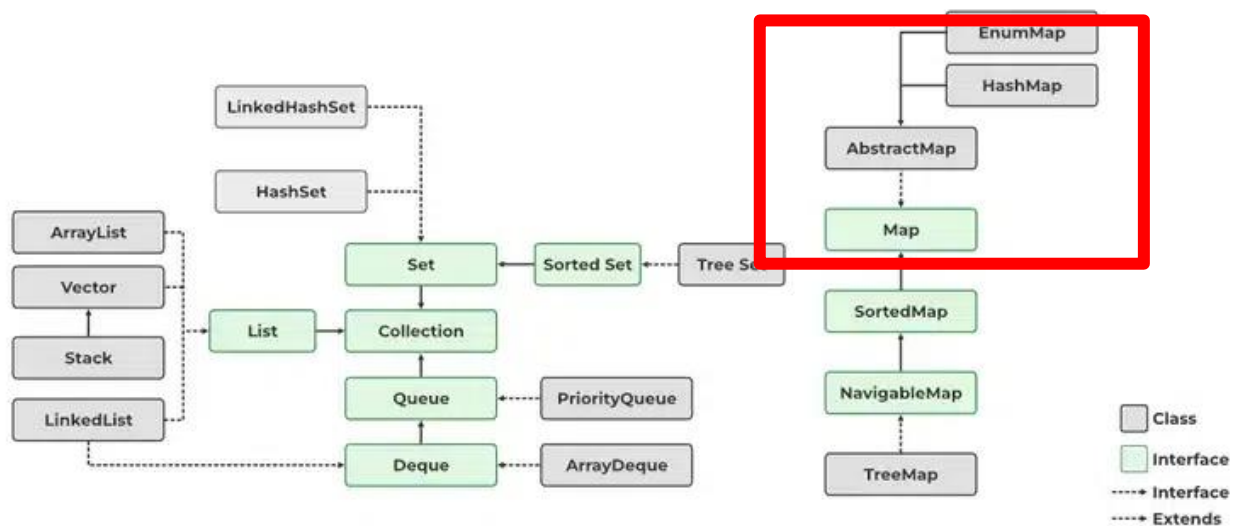
Questions :

1. Combien de méthodes abstraites sont déclarées dans l'interface « **List** » ?
2. Quelle méthode de l'interface « **List** » est à réimplémenter pour remplacer un élément à une position donnée ?
3. Quelle est différence entre la classe « **ArrayList** » et la classe « **Vector** » ?
4. Quelle structure de données représente la classe « **Stack** » ?

5. Quel est l'objectif de la méthode « **pop** » pour un objet de la classe « **Stack** » ?
6. Quelle opération permet d'ajouter un élément à un objet de la classe « **Stack** » ?
7. Qu'est la classe « **Stack** » par rapport à la classe « **Vector** » ?
8. Quels sont les avantages d'objets de la classe « **LinkedList** » par rapport à la classe « **ArrayList** » ?

2.1.2 Questions portant les classes qui implémentent l'interface « Map »

Le diagramme suivant indique les classes que vous allez étudier :



Information

Vous pouvez consulter la vidéo suivante pour une présentation du fonctionnement de la classe « HashMap » :

<https://www.youtube.com/watch?v=I9aBP0xm-IE>

Questions :

1. [Qu'est-ce qu'un tableau associatif \(ou « table associative », ou « dictionnaire »\) ?](#)
2. En Java, quelle classe peut être utilisée pour créer un tableau associatif ?

3. Lors de l'instanciation d'un objet de la classe « **HashMap** », à quoi correspondent les classes indiquées entre les chevrons ?

Par exemple, à quoi correspond « **<String, Integer>** » dans l'extrait suivant :

```
HashMap<String, Integer> mapTest = new HashMap<>();
```

4. Quelle méthode peut être utilisée pour ajouter une paire « **clef-valeur** » à un tableau associatif ?
5. Dans quelle classe (ou interface) est déclarée la méthode que vous avez trouvé en répondant à la question précédente ?
6. Quel est le type de retour de la méthode « **get(Object key)** » de la classe « **HashMap** » ?

2.2 IMPLEMENTATION D'EXEMPLE

Créez/trouvez un exemple concret d'utilisation des classes suivantes :

- « **Stack** »
- « **Hashmap** »

A faire

Pour chaque exemple d'utilisation, implémentez une fonction que vous ajouterez à un projet nommé « **api-collection-initiation** » sur un dépôt Git.

Il vous est également demandé d'ajouter les réponses aux questions précédentes dans un fichier ajouté à ce projet.

In case of fire



1. **git commit**



2. **git push**



3. **leave building**

CREDITS

ŒUVRE COLLECTIVE DE l'AFPA

Sous le pilotage de la DIIP et du centre d'ingénierie sectoriel Tertiaire-Services

Equipe de conception (IF, formateur, mediatiseur)

Michel Coulard – Formateur Evry

Chantal Perrachon – IF Neuilly sur Marne

Date de mise à jour : 03/07/2024

Reproduction interdite

Article L 122-4 du code de la propriété intellectuelle.

« Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droits ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction par un art ou un procédé quelconque. »