**BLOC2 ALGO.2 : Manipulation Objet - Collection - Dictionnaire**

|  |
| --- |
| Objets d’un traitement informatique  |

Introduction

L'objectif de ce chapitre et de gérer, manipuler les objets sous une autre forme que les tableaux. Il existe des bibliothèques de classe permettant de gérer les objets de manière dynamique.

**question 1** : Quel sont les inconvénients des tableaux ?

# Les Collections

Une collection représente un groupe d'objets, connu par ses éléments. Certaines collections acceptent les doublons, d'autres pas (En Algorithmie nous ferons abstraction de toutes contraintes). Certaines sont ordonnées, d'autres pas.

Il s'agit en quelque sorte d'une liste d'objet.

En Java il existe donc plusieurs types de collection. L'ensemble des collections sont des classes avec des propriétés et méthodes particulières

## En Algorithme

Voici à quoi ressemble la classe collection



**question 2** : Créer une fonction qui reçoit en paramètre 2 élèves et qui retourne une collection d'élève.

**question 3** : Ecrire une procédure qui reçoit en paramètre une collection d'élève et affiche le nombre d'élève

**question 4** : Ecrire une procédure qui reçoit en paramètre une collection d'élève et qui affiche pour chaque élève son nom et son prénom.

**question 5 :** Ecrire une procédure qui reçoit en paramètre une collection d'élève et affiche la date de naissance d'un nom, prénom passé en paramètre.

**question 6** : quels sont les inconvénients des collections (en s'appuyant sur la question 5)

## En Java (ArrayList)

ArrayList est un de ces objets qui n'ont pas de taille limite et qui, en plus, acceptent n'importe quel type de données, y compris NULL ! Nous pouvons mettre tout ce que nous voulons dans un ArrayList.

Exemple :



Il est préférable de "typer" une ArrayList.

Exemple :

ArrayList <Livre> al ;

al = **new** ArrayList<Livre>();

Sachez aussi qu'il existe tout un panel de méthodes fournies avec cet objet :



* size() : retourne le nombre d'élément
* add(int index, object element) : insère l'objet à l'index souhaité

**question 7** : Remplir le tableau ci dessous



**TD1** 

## en PHP



# Les dictionnaires

Un dictionnaire fonctionne avec un couple clé-valeur.

La clé, qui sert à identifier une entrée, est unique. La valeur, au contraire, peut être associée à plusieurs clés.

Contrairement aux collections, on n'insère pu à un index mais dans une clé !

## En Algorithme

Nous allons reprendre les exercices en 2.1 mais en mettant les élèves dans un dictionnaire.

**question 8 :** en s'inspirant de la déclaration de classe collection, écrire la classe dictionnaire avec ses méthodes

**question 9** : Créer une fonction qui reçoit en paramètre 2 élèves et qui retourne un dictionnaire d'élève.

* La clé de l'élève sera la première lettre de son prénom puis son nom de famille.
* La fonction GAUCHE(chaine,nb) permet de retourner les nb premiers caractères de la chaine.

**question 10** : Ecrire une procédure qui reçoit en paramètre un dictionnaire d'élève et affiche le nombre d'élève

**question 11** : Ecrire une procédure qui reçoit en paramètre un dictionnaire d'élève et qui affiche pour chaque élève son nom et son prénom (ainsi que sa clé)

**question 12 :** Ecrire une procédure qui reçoit en paramètre un dictionnaire d'élève et affiche la date de naissance d'un nom, prénom passé en paramètre.

**question 13** : quels sont les inconvénients & avantages des dictionnaires (en s'appuyant sur la question 12)

## En java (Hashtable)



Il est préférable de "typer" une Enumeration.

Exemple :

Enumeration <Livre> en;

en= ht.elements() ;

## En PHP