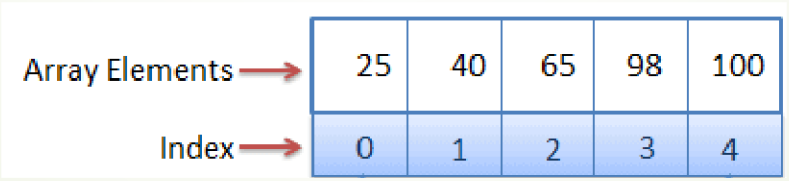
LES COLLECTIONS EN POO

Mr. CHAOULID

1. Les tableaux (Rappel)

- Un tableau (**Array**) est un ensemble d'éléments qui ont le même type et qui sont accessibles par leurs indices.
- Pour accéder à un tableau, on utilise l'indice compris entre 0 et la taille -1.



1. Les tableaux (Rappel)

- Les tableaux sont utiles, mail ils ont leurs limites par exemple :
  - Un tableau doit contenir des éléments de même type.
  - La gestion de la taille.
  - Ajout, la suppression

- Une alternative aux tableaux en POO est l'usage de **Collection**.

2. Les collections

- Une collection représente un regroupement d'objets ou de valeurs qui peuvent être de même type ou de type différent. c'est un objet dans lequel on peut stocker, organiser et manipuler des données de manière efficace et aisément à l'aide des méthodes prédéfinies.
- Une collection fonctionne plutôt comme un groupe d'éléments dans laquelle il est possible d'ajouter ou d'enlever un élément à n'importe quel endroit sans avoir à se préoccuper de la taille de la collection, ni où se trouve l'élément. Le nombre d'élément n'est pas défini au départ comme dans un tableau statique.

2. Les collections

- Les types de collections varient en fonction du langage de programmation, mais il existe certains concepts couramment utilisés. Voici quelques-uns des types de collections les plus courants en POO :

3. Types de collection

- **Listes :**
  - Les listes sont des collections ordonnées d'éléments.
  - Les éléments peuvent être de différents types.
  - L'accès aux éléments se fait par leur indice (position dans la liste).
  - **Exemple en Python :**

```
ma_liste = [1, 2, 3, 'quatre', 5]
```

3. Types de collection

- **Tuples :**
  - Les tuples sont similaires aux listes, mais ils sont immuables (on ne peut pas modifier leur contenu après leur création).
  - **Exemple en Python** :

```
mon_tuple = (1, 2, 3, 'quatre', 5)
```

3. Types de collection

- **Ensembles :**
  - Les ensembles sont des collections non ordonnées d'éléments uniques.
  - Ils sont souvent utilisés pour effectuer des opérations ensemblistes telles que l'union, l'intersection, etc.
  - **Exemple en Python** :

```
mon_ensemble = {1, 2, 3, 4, 5}
```

3. Types de collection

- **Dictionnaires :**
  - Les dictionnaires sont des collections associatives composées de paires clé-valeur.
  - Ils permettent d'accéder rapidement à une valeur en utilisant une clé.
  - Exemple en Python :

```
mon_dictionnaire = {'cle1': 'valeur1', 'cle2': 'valeur2', 'cle3': 'valeur3'}
```

3. Types de collection

- **Files (queues) :**
  - Les files sont des collections ordonnées qui suivent le principe du premier entré, premier sorti (FIFO).
  - Elles sont souvent utilisées pour la gestion de tâches en attente d'exécution.
  - **Exemple en Python** :

```
from collections import deque
ma_file = deque([1, 2, 3, 4, 5])
```

4. Les listes d'objet:

- En Python, les listes peuvent contenir n'importe quel type d'objet, y compris des objets de vos propres classes personnalisées.

- **Exemple:**

```
p1=Personne("Nom1","Prenom1",10)
p2=Personne("Nom2","Prenom2",20)
p3=Personne("Nom3","Prenom3",30)
L=[p1,p2,p3]

#L=[Personne("Nom1","Prenom1",10),Personne("Nom2","Prenom2",20),Personne("Nom
3","Prenom3",30)]

for p in L:
    print(p)
```

TP6

Mr. CHAOULID