Université Tunis El-Manar	Faculté des Sciences de Tunis
Module : Programmation.O.O	**Année universitaire** : 2024-2025
Section : MI 2 – PI 2	Hajer Dammak

Série N° 2 : Les constructeurs et l'encapsulation en Java

Points essentiels :

- Constructeur
- Encapsulation
- La référence `this`

Exercice 1 : Un point

1) Définir une classe nommée `Point` comportant :
   - Deux attributs `x` et `y` de type `double` représentant les coordonnées d'un point.
   - Un constructeur recevant en arguments les deux valeurs des coordonnées d'un point à construire.
   - Une méthode nommée `affiche` dont le rôle est d'afficher sur écran les coordonnées d'un point sous la forme : `p=(x,y)`.
   - Une méthode nommée `translate` recevant en arguments deux valeurs doubles `dx` et `dy` et dont le rôle est de translater un point de `dx` et `dy`.
2) Ecrire une méthode main (à mettre dans la classe `Point` ou dans une autre classe `TestPoint`) qui :
   - Construit un point de coordonnée 2.5 et 7.
   - Affiche les coordonnées de ce point.
   - Translate ce point de 2 et de 3.
   - Réaffiche les coordonnées du point

Exercice 2 : Un rectangle

1) Définir une classe nommée `Rectangle` comportant deux attributs `longueur` et `largeur` de type `float` et les méthodes suivantes :
   - `Rectangle(float longueur, float largeur)` : constructeur recevant comme paramètres la longueur et la largeur.
   - `float permimetre()` : méthode qui retourne le périmètre d'un rectangle `(longuer+largeur) *2`.
   - `float surface()` : méthode qui retourne la surface d'un rectangle `longuer*largeur`.
   - `void affichage()` : méthode qui affiche les dimensions d'un rectangle, son périmètre et son surface.
   - `public static void main(String[] args)` : méthode principale dans laquelle on crée et on affiche toutes les informations sur un rectangle de longueur 7.5 et largeur 3.8.
2) Ajouter dans cette classe un deuxième constructeur sans paramètres : `Rectangle()` qui permet d'initialiser les attributs de la classe `Rectangle` par les valeurs 12 et 7. Compléter la méthode `main` par la création d'un second rectangle en utilisant ce deuxième constructeur.

3) Enlever le premier constructeur et ajouter dans Rectangle la méthode void modifierRectangle(float lon, float lar) dont le rôle est de changer les dimensions d'un rectangle par les valeurs de ses deux paramètres. Créer ensuite main un rectangle dont les dimensions sont 16.3 et 15.
4) a. Si on enlève tous constructeurs de la classe Rectangle qu'obtient-on à l'exécution en écrivant dans main les instructions suivantes :

```
Rectangle r = new Rectangle() ; r.affiche() ;
```

b. Peut-on écrire les instructions précédents dans main si on avait gardé le premier constructeur Rectangle(float longueur, float largeur)?

Exercice 3 : Une Date

1) Définir une classe nommée Date qui permet de représenter le format de date suivant : 07/10/2023. Cette classe doit contenir les méthodes suivantes :
   - nombreJours() : donne le nombre de jours pour le mois d'une date.
   - dateValide() : permet de vérifier si une date est valide.
   - lendemain() : donne la date de demain.
2) Ecrire la méthode main dans une classe TesterDate qui permet de :
   - Créer une date à partir des valeurs jour, mois et annee introduits à partir de la ligne de commande.
   - Afficher cette date sous le format décrit en haut jj/mm/aaaa.
   - **Si** la date est valide : a) affiche le nombre de jours pour du mois de cette date.
     b) affiche la date de demain.
   - **Sinon** affiche un message d'erreur.

Remarque :
a. Respecter le principe de l'encapsulation en déclarant les attributs et les méthodes.
b. Compléter la classe Date par d'autres méthodes si c'est nécessaire.

Exercice 4 : Compte Bancaire

Créer une classe Compte qui modélise un compte en banque. Dans cette classe on a les informations suivantes : numéro du compte, le solde du compte et le client associé.
Cette classe doit posséder les méthodes suivantes :

- Constructeur avec paramètres.
- getSolde() pour consulter le solde d'un client donné par son numéro de compte.
- autoriser() pour vérifier si un client est autorisé ou non à retirer un montant donné en paramètre.
- Deposer() pour déposer un montant donné au profit d'un compte.
- Retirer() pour retirer un montant donné à partir d'un compte.
- Afficher() pour afficher les informations d'un compte.

Un client est une Personne ayant un nom, prénom, un mot de passe et un tableau de comptes. Pour une personne, on peut :

- Consulter le nom par la méthode getNom().

- Consulter le prénom par la méthode `getPrenom()`.
- Vérifier si un mot de passe est correct ou non.
- Ajouter un compte au tableau des comptes si c'est possible.
- Afficher les informations d'une personne et son nombre de comptes.

1) 1- Implémenter les deux classes Personne et Compte avec les constructeurs suivants :
   - `Personne(String, String, String)`
   - `Compte(String, float, Personne)`
2) Implémenter les méthodes nécessaires des deux classes `Personne` et `Compte`.
3) Implémenter une classe de teste qui teste les différentes classes en passant par les étapes suivantes :
   - Créer une personne `p`,
   - Créer deux comptes `c1` et `c2`,
   - Ajouter les deux comptes `c1` et `c2` à la personne `p`,
   - Afficher les informations de la personne `p`,
   - Déposer la somme de 100 dinars dans les comptes `c1` et `c2`,
   - Retirer la somme de 50 dinars du compte `c1`,
   - Afficher les nouvelles informations des comptes `c1` et `c2`,
   - Afficher les nouvelles informations de la personne `p`.

Exercice 5 : Temps

1) Créer la classe `Temps` qui contient 3 attributs entiers privés : `heure`, `minute`, `seconde`.
2) Définir les constructeurs suivants en utilisant chaque fois que c'est possible la référence `this`.
   - `Temps (int heure)` : construit un objet Temps à partir d'une heure donnée
   - `Temps (int heure, int minute)` : construit un objet Temps à partir des deux valeurs heure et minute.
   - `Temps (int heure, int minute, int seconde)` : construit un objet Temps à partir des trois valeurs heure, minute, seconde.
3) Définir une méthode affiche () qui affiche le temps sous la forme suivante :

   `"Il est ... heures ... minutes ... secondes"`

4) Qu'obtient-on à l'exécution si on ajoute dans cette classe la méthode main suivante :

```
public static void main(String[] args)
{
   Temps t = new Temps (10); t.affiche();
   t = new Temps (10, 12); t.affiche ();
   t = new Temps (10, 12, 45); t.affiche ();
}
```