

DEVOIR SEMESTRE 1
Matière : INFORMATIQUE
Classes : 2^{ème} Année Durée : 1 h
Préparation : MP-PC-PT



Problème 1 (14 pts)

L'objectif est de créer un système de gestion de magasin en utilisant la programmation orientée objet en Python. Le système devrait permettre aux utilisateurs de gérer le stock des produits et les ventes dans un magasin en utilisant deux classes : **Produit** et **Magasin**.

La classe **Produit** contient les attributs **identifiant**, **nom**, **prix** et **stocks**.

La classe **Magasin** contient un attribut **stock** contenant la liste de produits en stock du magasin.

Travail demandé : Écrire un programme Python permettant de :

1. Définir le **constructeur** de la classe **Produit** qui prend en paramètre l'identifiant, nom, prix et stocks

Exemple : `produit1 = Produit(1, "Ordinateur portable", 800, 10)`

`produit2 = Produit(3, "Téléphone portable", 400, 20)`

2. Définir la méthode **ajouter_stock(...)** qui prend en paramètre la quantité à ajouter au produit en stocks.

Exemple : `produit1.ajouter_stock(20)`

3. Définir la méthode **retirer_stock(...)** qui prend en paramètre la quantité à retirer du produit en stocks et **retourne** un message selon le cas. Dans le cas où cette quantité n'est pas disponible, retourner un message indiquant "Stock insuffisant de nom du produit". Dans le cas contraire retourne "quantité (retirée) nom du produit (s) retiré(s) du stock"

Exemple : `produit1.retirer_stock(35)` retourne "Stock insuffisant de **Ordinateur portable**".

Dans le cas contraire, `produit1.retirer_stock (25)` retourne "**25 Ordinateur portable(s)** retiré(s) du stock".

4. Définir la méthode **afficher_info(...)** permettant de retourner les informations sur le produit.

Exemple : lorsqu'on a `produit1 = Produit(1, "Ordinateur portable", 800, 10)`

`produit1.ajouter_stock(20)`

`produit1.retirer_stock (25)` on aura

"Produit : **Ordinateur portable**, Prix : **800** Dt, Stocks disponible : **5**"

5. Définir le **constructeur** de la classe **Magasin** permettant d'initialiser **stock** en une liste vide.

Exemple : `magasin = Magasin()`

6. Définir la méthode **ajouter_produit(...)** qui prend en paramètre le produit à ajouter dans la liste **stock**.

```
magasin.ajouter_produit(produit1)
```

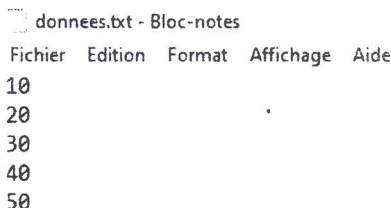
```
magasin.ajouter_produit(produit2)
```

7. Définir la méthode **retirer_produit(...)** qui prend en paramètre le produit à retirer de la liste **stock**.

Exemple : `magasin.retirer_produit(produit1)`

Problème 2 (6 pts)

Le programme doit être capable de lire des données numériques à partir d'un fichier texte donné « `donnees.txt` » représenté dans la figure 1. Le fichier de données doit contenir une valeur par ligne. Les données doivent être stockées dans une structure de données appropriée (par exemple, une liste) pour être utilisées dans le tracé de courbe. Utilisez la bibliothèque **matplotlib** pour tracer les données sous forme de courbe.



```
donnees.txt - Bloc-notes
Fichier Edition Format Affichage Aide
10
20
30
40
50
```

Figure 1. Fichier : `donnees.txt`

Travail demandé : Écrire un programme Python permettant de :

1. Initialiser une **liste donnees** vide. Puis la remplir par les valeurs à partir de votre fichier. A l'exécution on obtient : `[10.0, 20.0, 30.0, 40.0, 50.0]`
2. Créer une **liste indices** pour les points de données : commence par 1 jusqu'à la longueur de la liste des `donnees`. A l'exécution on aura : `indices= [1, 2, 3, 4, 5]`
3. Tracer la courbe **indices** en fonction de **donnees**

ANNEXE – Quelques Fonctions/Méthodes Python

- **lst.append(val)** ajoute val à la fin de la liste lst
- **lst.remove(val)** supprime la première occurrence val de la liste lst
- **st.strip()** Retourne une copie de st en supprimant à la fois les retours à la ligne de fin et de début
- **len(lst)** : retourne le nombre d'éléments de la liste lst.
- **plot(abscisses, ordonnées)** permet de tracer des courbes qui relient des points dont les abscisses et ordonnées sont fournies en arguments.
- **show()** fonction du module pyplot de la bibliothèque matplotlib est utilisée pour afficher toutes les figures.

