Feuille d'exercices N°2

(Agrégation et délégation)

Exercice 1 *Adresse*

On considère une entité `Adresse` décrite par un numéro, un nom de rue, un code postale et une ville.

Q.1 Donnez une implémentation de l'entité `Adresse`, en donnant la possibilité de retourner ou de modifier n'importe quel champ de l'entité par des méthodes définies dans la classe, prévoir une méthode d'affichage.

Q.2 Créer une entité `Citoyen` décrite par un nom, prénom et adresse. Pour les méthodes, on demande de définir uniquement le constructeur et une méthode d'affichage.

Q.3 Ecrire un programme qui crée un tableau `C` de `10` citoyens et affiche ensuite tous les citoyens `(nom, prénom, adresse)` habitant dans une même ville `v` donnée.

Exercice 2 *Répertoire*

On considère une entité `Repertoire` décrite par un nom, un nombre de fichiers contenus, une taille en nombre d'octets et une date de création du répertoire.

Q.1 Donnez la description d'une entité `Date` en incluant une méthode `apres(Date d)` qui vérifie si une date est postérieure à une autre.

Q.2 Donnez la description de l'entité `Repertoire`, en donnant la possibilité de retourner ou modifier n'importe quel champ de l'entité par des méthodes définies dans la classe, prévoir une méthode d'affichage.

Q.3 Ecrire un programme qui crée un tableau `R` de 20 objets `Repertoire` et affiche tous les répertoires vides créés avant une date `d` donnée.

Exercice 3 *Message*

On considère une entité `Message` décrite par un identifiant, un expéditeur, un destinataire, un contenu (chaines de caractères) et une date d'envoi.

Q.1 Donnez la description d'une entité `Date` en incluant une méthode `egal(Date d)` qui vérifie l'égalité entre deux dates.

Q.2 Donnez la description de l'entité `Message`, en donnant la possibilité de retourner ou modifier n'importe quel champ de l'entité par des méthodes définies dans la classe, prévoir une méthode d'affichage.

Q.3 Ecrire le programme qui crée un tableau `M` de 20 messages et affiche tous les messages dont la date d'envoi est égale à une date `d` donnée.

Exercice 4 *Un problème de configuration*

L'objectif de cet exercice est de maîtriser les concepts d'**agrégation** et de **délégation**.

Compétences à acquérir :
- écrire les constructeurs d'une classe agrégeant des objets : construire mais aussi faire construire les objets composites,
- définir des accesseurs pour la nouvelle classe ainsi que des méthodes par délégation.

Sujet :
On souhaite concevoir un outil d'aide à la gestion des commandes d'un fournisseur de produits informatiques. Les classes `Client`, `ComposantMateriel` et `ComposantLogiciel` sont fournies en annexe.

On vous demande d'écrire la classe `Configuration` qui modélise une configuration livrée à un client. Une configuration est constituée d'un composant matériel (un ordinateur), d'un composant logiciel (système d'exploitation et logiciels installés sur la machine), et on lui associe un propriétaire.

```
public class Configuration {
    private Client proprietaire;
    private ComposantMateriel materiel;
    private ComposantLogiciel logiciel;
    /* à completer */
}
```

A. Constructeur et accesseurs :

Ecrire le corps du constructeur et des accesseurs.

```
/* Constructeur */
public Configuration(Client proprietaire,
        ComposantMateriel materiel,
        ComposantLogiciel logiciel) {
/* à completer */
}
/* Accesseurs */
public double getPrix() {
/* à completer */
}
public double getFrequence() {
/* à completer */
}
public int getMemoire() {
/* à completer */
}
public int getDisqueDur() {
/* à completer */
}
public void setFrequence(double frequence) {
/* à completer */
```

```
}
public void setMemoire(int memoire) {
/* à completer */
}
public void setDisqueDur(int disqueDur) {
/* à completer */
}
```

B. Evaluation du prix d'une modification matérielle :

Ecrivez la méthode `prixModification` calculant le prix d'une modification de la configuration courante (modification de la fréquence du processeur, de la capacité de la mémoire, de la capacité du disque dur ou encore du composant logiciel). Le prix de la modification est égal à la différence entre le prix de la nouvelle configuration et le prix de la configuration courante.

```
/* Autres methodes d'instances */
public double prixModification(double frequence,
        int memoire,
        int disqueDur,
        ComposantLogiciel logiciel) {
/* à completer */
}
```

C. Programme principal :

Ecrivez la méthode `toString()` de la classe `Configuration`. Ecrivez ensuite un programme principal testant votre implémentation en créant une configuration pour "Kateb", qui habite au "2, rue Larbi Ben M'Hidi 16000 Alger" composée d'un microprocesseur ayant une fréquence de 2.8 Ghz, 4 Go de mémoire, un disque dur de 500 Go, "Linux" pour système d'exploitation et aucun pack logiciel. Votre programme principal affichera la configuration, son prix, ainsi que le prix de la modification consistant à doubler les capacités de la mémoire et du disque dur.

```
public String toString() {
/* à completer */
}
/* Methode de classe pour tester */
public static void main(String[] arg) {
/* à completer */
}
```

D. **Annexe** :

```
public class Client {
   // constructeur
   public Client(String nom, String adresse)
   // accesseurs
   public String getNom()
   public String getAdresse()
   public void setNom(String nom)
   public void setAdresse(String adresse)
   // autres méthodes d'instance
   public String toString()
}
```

```
public class ComposantLogiciel {
   // constructeurs
   public ComposantLogiciel(String OS, boolean packOffice,
        boolean packAdditionnel)
   public ComposantLogiciel(ComposantLogiciel l)
   // accesseurs
   public double getPrix()
   // autres méthodes d'instance
   public String toString()
}
```

```
   public class ComposantMateriel {
   // constructeur
   public ComposantMateriel(double frequence, int memoire,
        int disqueDur)
   // accesseurs
   public double getPrix()
   public double getFrequence()
   public int getMemoire()
   public int getDisqueDur()
   public void setFrequence(double frequence)
   public void setMemoire(int memoire)
   public void setDisqueDur(int disqueDur)
   // autres méthodes d'instance
   public String toString()
}
```

Bon travail !