

TD 2

Exercice 1

Un fichier est caractérisé par : son nom, son extension, son emplacement et sa taille sur disque (en kilo octets).

Les fichiers sont enregistrés dans des supports magnétiques pouvant être des disquettes ou des disques durs

Une disquette est caractérisée par son nom logique, la taille d'un secteur en kilo octets, le nombre de secteurs par pistes et le nombre de pistes par surface.

Un disque dur est caractérisé par son nom logique, la taille d'un secteur en kilo octets, le nombre de secteurs par piste, le nombre de pistes par surfaces et le nombre de surfaces.

Afin de pouvoir gérer son contenu, on suppose que chaque disque contient deux tableaux

- Le premier contient toutes les caractéristiques d'un fichier donné
- Le second contient pour chaque fichier son adresse physique (les adresses physiques sont attribuées automatiquement par le système)

On se propose alors de développer les classes **Fichier**, **Disquette** et **DisqueDur** :

La classe **Fichier** doit contenir :

- Un constructeur paramétré qui accepte comme arguments une liste de valeurs permettant l'initialisation des différents attributs de l'objet en question;
- Des mutateurs permettant la modification des valeurs des attributs d'un objet;
- Des accesseurs permettant le renvoi des valeurs des attributs d'un objet;

Chacune des deux classes **Disquette** et **DisqueDur** doit contenir :

- Un constructeur paramétré qui accepte comme arguments une liste de valeurs permettant l'initialisation des différents attributs de l'objet en question;
- Une méthode **getEspaceDisque()** permettant de calculer et de retourner la taille totale du disque concerné (en kilo octets)
- Une méthode **getEspaceLibre()** permettant de calculer et de retourner l'espace libre dans un disque (en kilo octets)
- Une méthode **ajouterFichier()** permettant d'ajouter un fichier donné (ajouter un fichier revient tout simplement à l'insérer à la fin du tableau si la taille de disque suffit, on s'occupe pas de l'adresse ou il sera stocké)
- Une méthode **explorerExtension()** permettant d'afficher tous les fichiers (nom, extension, emplacement et taille) ayant comme extension celle qui est introduite comme paramètre.

Dans le but d'expérimenter le comportement de toutes les classes déjà développées on se propose d'ajouter une nouvelle classe nommée **Application** basée principalement sur la méthode **main ()**. Dans cette méthode principale on doit :

- Construire les fichiers suivants :
 - (Nom : " examen " ; Extension : " doc " ; Taille 2 kilo ; Emplacement : " C:\word ")
 - (Nom : " bulletin " ; Extension : " xls " ; Taille 1 kilo ; Emplacement : " C:\excel ")
 - (Nom : " nature " ; Extension : " gif " ; Taille 25 kilo ; Emplacement : " C : ")
- construire le disque dur suivant :

(Nom logique : " C : " ; Taille d'un secteur : 0.9375 kilo octet ; Nombre de surfaces: 16 ; Nombre de piste par surface : 1024 ; Nombre de secteurs par pistes : 520)
- ajouter tous les fichiers créés à ce disque
- afficher les noms des fichiers ayant comme extension « doc ».

Exercice 2

Nous allons programmer le calcul des salaires hebdomadaires des employés d'une entreprise.

Cette entreprise comporte plusieurs types d'employés :

- Des employés qui sont payés suivant le nombre d'heures qu'ils ont travaillées dans la semaine. Ils sont payés à un certain tarif horaire et leurs heures supplémentaires (au-delà de 39 heures) sont payées 30 % de plus que les heures normales.
- D'autres employés, payés de la même façon, mais leurs heures supplémentaires sont payées 50 % de plus que les heures normales.
- Les commerciaux sont payés avec une somme fixe à laquelle on ajoute 1 % du chiffre d'affaires qu'ils ont fait dans la semaine.

1) Créer 3 classes dérivées (employé_ordinaire, employé_responsable et commerciaux)et une classe de base employé

- Vous donnerez un nom à chacun des employés. On ne pourra modifier le nom d'un employé.
- Vous commencerez par écrire une classe Employe dont hériteront les autres classes.
- Pour simplifier, vous ne donnerez qu'un seul modificateur setInfosSalaire pour entrer ou modifier les informations nécessaires au calcul des salaires.
- Les classes comporteront au moins 2 constructeurs : un qui ne prend en paramètre que le nom de l'employé et l'autre qui prend en paramètres le nom et toutes les informations pour le calcul du salaire.

Une classe **Paie** comportera une unique méthode main qui entrera les informations sur des employés des différents types (au moins 3 commerciaux). Les employés seront enregistrés dans un tableau **employes**.

Au moins un des employés sera créé avec le constructeur qui n'a que le nom en paramètre, et vous entrerez ensuite les informations pour son salaire avec la méthode **setInfosSalaire**. Pour au moins un autre employé, vous utiliserez le constructeur pour entrer les informations sur le salaire.

La méthode ***main*** affichera le salaire hebdomadaire de chacun des employés dans une boucle "***for***" qui parcourra le tableau des employés. Vous utiliserez le polymorphisme avec un accesseur pour le salaire. L'affichage aura exactement la forme : "**Mohamed gagne 700 D**".

Travail à faire

- a – implémentez les différentes classes
- b- écrivez la classe paie
- c- donnez l'affichage de l'exécution de la classe main