

Programmation Orientée Objets

Transtypage

E. Hyon L. Mesnager

2022-2023

Ce TD a pour but la manipulation et le transtypage des objets

Exercice 1 :

On cherche à modéliser les différentes manières de procéder en fonction du type d'ordinateur que l'on manipule. On suppose qu'on manipule trois classes d'ordinateurs : les ordinateurs, les portables et les tablettes. Lisez l'énoncé jusqu'au bout pour déterminer les attributs et affiner les héritages.

Question 1

Écrire les trois classes et les constructeurs associés. On suppose que les portables sont des ordinateurs particuliers et que les tablettes sont des portables de type particulier.

Question 2

Écrire une méthode `identite` qui affiche le type d'ordinateur (par ex : *je suis un portable*) sans qu'on utilise d'attribut spécifique.

Question 3

Écrire une méthode qui affiche le numéro de série.

On considère qu'un ordinateur utilise un clavier externe uniquement. La manière dont sont saisies les données dépend des ordinateurs (clavier externe pour les ordi, clavier interne pour les portables, clavier tactile pour les tablettes). Mais les portables et les tablettes ont la possibilité de connecter un clavier externe.

Question 4

Écrivez la méthode `saisie` qui affiche la manière dont sont saisies les données (clavier externe, clavier interne, tactile) sauf si il y a un clavier connecté pour un portable ou une tablette et dans ce cas, on fait appel à la méthode de la super classe.

Question 5

Écrire un main dans une classe `Principale` qui crée 7 ordi (2 tours, 3 portables et 2 tablettes) et qui affiche les numéros de série. Tous les ordinateurs sont stockés dans un tableau. Faut-il créer des tableaux différents pour stocker les portables et les tablettes ?

Question 6

Ecrire dans la classe `Principale` une méthode qui affiche toutes les identités des ordinateurs créés.

Chaque ordinateur a un module de connections qui décrit ses différentes interfaces de connections disponibles : Ethernet pour tous, wifi pour les portables et les tablettes et 3G pour les tablettes. Quel est le choix d'implémentation le plus pertinent pour modéliser cela ?

Question 7

On crée une **méthode d'instance** d'Ordinateur `etherAdHoc()` qui prend en paramètre un ordinateur et qui affiche si il peut être relié à l'instance par ethernet (il faut pour cela que son interface ethernet soit activée).

Question 8

On crée une **méthode d'instance** de portable `WifiAdHoc()` qui prend en paramètre un ordinateur et qui affiche si il peut être relié à l'instance par wifi (il faut pour cela que son interface wifi existe et qu'elle soit activée).

Question 9

On crée une **méthode d'instance** de tablette `3GAdHoc()` qui fait la même chose que la méthode du dessus pour la 3G.

Question 10

Ecrire une méthode d'instance `AdHoc` qui prend un ordinateur en paramètre et qui affiche s'il est possible de se connecter en mode Ad Hoc avec lui. Une réponse doit être fournie quelque soit le type d'ordinateur. On ne doit pas utiliser `instance of` mais redéfinir la méthode selon les classes.

Question 11

Ecrire dans la classe principale la méthode qui affiche les ordinateurs pouvant être reliés entre eux par une méthode `adHoc`.

Question 12

Ecrire la méthode `mobilité` qui ne s'applique qu'aux portables et tablettes et qui affiche *Je suis mobile*.

Question 13

Ecrire dans la classe principale la méthode qui affiche les ordinateurs qui sont mobiles.

*Optionnelles***Question 14**

- Ecrire une méthode d'instance `ports(int x)` pour la classe `Ordinateur` qui affiche le nombre x .
- Ecrire une méthode d'instance `ports(int x, int y)` pour la classe `Portable` qui affiche le nombre $x + y$.
- Ecrire une méthode d'instance `ports(int x, int y, int z)` pour la classe `Portable` qui affiche le nombre $x + y + z$.

Question 15

Parcourez votre tableau en appliquant à tous les ordinateurs lorsque c'est possible la méthode 1, la méthode 2 et la méthode 3. Vous ferez des downcasting si nécessaire.

Question 16

Ecrire la méthode `PuissanceSignal()` qui prend pour paramètres un portable et un float pour la classe `portable` et une tablette et un entier dans la classe `tablette`. Elle affiche le nombre et le type sous la forme “ *La force du signal est de X pour ce (portable ou tablette) qui est du type* ”. (Le type est obtenu par la méthode `getClass()`).

Question 17

Lancer dans le main la méthode `PuissanceSignal()` avec toutes les combinaisons de paramètres possibles. Expliquez les raisons qui font que cela ne marche pas.

Question 18

On veut écrire une méthode de classe `connectionsListe()` qui prend en paramètre un ordinateur et qui affiche toutes les manières de se connecter possibles de l'ordinateur donné en paramètre en fonction du type.

Exercice 2 : Un peu de sport

On souhaite écrire une classe `Athlete` modélisant un athlète sportif. On décide que les seules informations enregistrées sont :

- son identité (une chaîne de caractères),
- sa nationalité (codée par un entier)
- et son niveau de performance (un nombre décimal).

Question 1

Ecrire la déclaration de la classe et de ses attributs. On décide que les attributs ont une visibilité paquetage. Par quel mécanisme une instance d'une autre classe peut-elle accéder à la valeur de ces attributs ?

Question 2

Définir le constructeur initialisant tous les attributs de la classe.

Question 3

On décide que deux instances de la classe `Athlete` sont dites égales si elles correspondent au même athlète (identité et nationalité identiques) et différentes dans le cas contraire. On souhaite munir la classe d'une méthode permettant de décider si deux instances sont égales ou différentes. On décide pour cela de redéfinir la méthode `equals` héritée de la classe `Object` dont la signature est

```
boolean equals(Object o)
```

Exercice 3 : Entier

Soit un chiffre présent sous une chaîne de caractères.

Question 1

Transformer ce mot en entier. Modifiez le puis affichez le. Faites la même chose dans une méthode.

Exercice 4 : Liste chaînée

Ecrire la classe `MaListeChaine` qui implémente une liste simplement chaînée d'habitations (voir plus bas). L'objet `Habitation` n'a besoin ici que d'être déclaré avec un constructeur sans paramètres.

Votre élément `Maillon` sera une classe privée déclarée dans votre classe `MaListeChaine`

On rappelle les fonctionnalités qu'une liste chaînée doit posséder

Elle peut être parcourue.

On peut insérer un élément au début.

De même un élément peut être supprimé à la place *i*.

Question 1

Ecrivez votre classe liste chaînée.

Exercice 5 : Impôts Locaux

Dans le cadre de l'informatisation d'une mairie, on veut automatiser le calcul des impôts locaux. On distingue deux catégories d'habitation : les habitations à usage professionnel et les maisons individuelles, l'impôt se calculant différemment selon le type d'habitation. Pour cela, on définit les classes `HabitationProfessionnelle` et `HabitationIndividuelle` et les caractéristiques communes à ces deux classes sont regroupées dans la classe `Habitation`.

La classe `Habitation` comprend les attributs : `proprietaire` du type chaîne de caractères et qui correspond au nom du propriétaire, `adresse` du type chaîne de caractères et qui correspond à l'adresse de l'habitation, `surface` du type `double` et qui correspond à la surface de l'habitation et qui permet de calculer le montant de l'impôt. Elle possède aussi les méthodes :

- `double impot()` qui permet de calculer le montant de l'impôt que doit payer le propriétaire de l'habitation à raison de 2 euros par m².
- `void affiche()` qui permet d'afficher les trois attributs de la classe `Habitation`.

et un constructeur à trois paramètres permettant d'initialiser une instance de la classe `Habitation` :
`Habitation(String P, String A, double S);`

Question 1

Définissez la classe `Habitation`.

Définition des classes : `HabitationIndividuelle` et `HabitationProfessionnelle`

Le calcul de l'impôt d'une maison individuelle est différent de celui d'une habitation, il se calcule en fonction de la surface habitable, du nombre de pièces et de la présence ou non d'une piscine. On compte 15 euros/pièce et 80 euros supplémentaire en cas de présence d'une piscine auxquels s'ajoute le prix dû à la surface.

Le calcul de l'impôt d'une habitation à usage professionnel est également différent de celui d'une habitation. Il se calcule en fonction de la surface occupée par le bâtiment et du nombre d'employés travaillant dans l'entreprise. On compte 150 euros supplémentaires par tranche de 10 employés.

Question 2

Définir les deux classes `HabitationIndividuelle` et `HabitationProfessionnelle`. On redéfinira si nécessaire les méthodes `Impot()` et `Affiche()` (qui permet d'afficher toutes les caractéristiques de l'habitation).

On veut gérer une application de suivi de paiement des impôts d'une ville. Pour cela on aura deux listes : celle des habitations ayant payés et celle des habitations devant encore payer.

Question 3

Pour cela créer (on va faire que le début de l'application) une liste chaînée dans laquelle on insère deux `HabitationIndividuelle` et deux `HabitationProfessionnelle`