

TP - Séance n°4

Héritage, Modélisation

Important : Dans ces trois exercices toutes les variables (d'objet ou de classe) seront **privées**.

Exercice 1 (Héritage et polymorphisme de base)

Rappel (opérateur instanceof) :

L'expression "`x instanceof nomClasse`" (resp. "`x instanceof nomInterface`") vaut `true` si la variable `x` contient un objet de type `nomClasse` ou d'une de ses sous-classes au sens large (resp. la variable `x` contient un objet dont le type implémente l'interface `nomInterface`).

1. Écrivez une classe `Batiment` avec deux variables `adresse` et `surfaceHabitable` (un entier) et son constructeur `Batiment(String adresse, double surface)`. Implémentez la méthode `String toString()`.
2. Écrivez une classe `Maison` héritant de `Batiment` avec les variables `nbPieces` et `surfaceJardin`. Écrivez le constructeur `Maison(String adresse, int surfaceH, int surfaceJ, int nbPieces)` en utilisant un appel à `super(...)`. Écrivez aussi la méthode `String toString()`.
3. Écrivez une classe `Immeuble` héritant de `Batiment` avec la variable `nbAppart`. Écrivez le constructeur correspondant et la méthode `String toString()`.
4. Écrivez une méthode `main` dans une classe `TestBatiment`. Ce programme doit instancier un bâtiment, une maison, un immeuble et les afficher. Ensuite, créez un tableau de 10 bâtiments. Est-ce que les bâtiments sont instanciés ? Remplacez 2 éléments du tableau par la maison et l'immeuble précédemment créés. Affichez tout le tableau.
5. On va maintenant implémenter les méthodes `getSurfaceHabitable()` ainsi que `getSurfaceJardin()`. Dans quelles classes les implémente-t'on ? Écrivez une méthode `surfaceHabitableTotale(Batiment[] tabBat)` dans la classe `TestBatiment` ainsi qu'une méthode `surfaceJardinTotale(Batiment[] tabBat)` qui prend en argument un tableau de bâtiments (avec éventuellement des cases vides) et calcule la surface totale habitable des bâtiments (resp. la surface totale des jardins).

Qu'ont de particulier ces deux méthodes ?

Pensez à utiliser `instanceof`.

Testez.

6. Dans l'île de Javuta, l'impôt local d'un bâtiment est calculé selon la formule

$$\text{Impot} = \text{TauxA} \times \text{surfaceHabitable} + \text{TauxB} \times \text{surfaceJardin}$$

Les valeurs de cette année étant $\text{TauxA} = 5.6$ et $\text{TauxB} = 1.5$.

Où déclarer les variables `TauxA` et `TauxB` ? et comment ? Dans quelle(s) classe(s) faut-il écrire la méthode `impot` ?

Exercice 2 1. Écrivez une classe `personne`. Une personne a un nom et un prénom.

2. Toujours dans l'île de Javuta, tous les bâtiments ont un propriétaire. Par contre, seuls les appartements des immeubles peuvent être loués, mais ce n'est pas automatique. On suppose que les appartements sont numérotés à partir de 1.

Modifiez vos classes pour qu'on puisse trouver le propriétaire d'un bâtiment, le locataire de l'appartement numéro n d'un immeuble, les différents biens immobiliers dont est propriétaire ou locataire une personne. . .

3. Quelles méthodes peut-on logiquement ajouter dans ces classes ?

Exercice 3 *Gestion d'un magasin d'instruments* Nous allons nous pencher dans cet exercice sur un programme permettant de gérer un magasin d'instruments de musique. Une classe `Magasin` sera au cœur de votre programme. Pour l'instant, contentez-vous de créer le fichier `Magasin.java` qui vous permettra de faire vos tests.

Le magasin gère les instruments suivants :

- Instruments à cordes
 - Pianos : à queue, droits, numériques ;
 - Guitares : acoustiques, électriques.
- Instruments à vent
 - Hautbois
 - Clarinette
 - Flûte traversière

Chaque instrument possède un nom, ainsi qu'un prix d'achat et un prix de vente. Les instruments à cordes possèdent une longueur et une largeur, tandis que les instruments à vent sont dotés d'un type : à biseau (flûte), à anche simple (clarinette), à anche double (hautbois).

Tous les pianos possèdent un nombre de touches (en général, 88). Les pianos numériques possèdent (ou non) le *toucher lourd*, ainsi qu'une qualité d'échantillonnage (allant de 1 à 3). Les pianos droits possèdent des cordes droites ou croisées, et sont dotés (ou non) d'un silencieux. Enfin, les pianos à queue ont de particulier leur taille, qui est notée "4/4", "1/2", "1/4", ou "Crapaud".

Les guitares acoustiques sont pour gaucher/droitier, et ont un tirant symbolisé par une valeur de 1 à 3. Les guitares électriques sont fournies (ou non) avec un amplificateur, et avec (ou non) des pédales d'effets sonores.

Une flûte traversière ou un hautbois gère ou non les trilles, et possède une qualité acoustique allant de 1 à 5. Une clarinette est adaptée à une utilisation symphonique, de chambre, ou harmonique.

Concevez une architecture orientée objet pour gérer les différents instruments de musique.

Écrivez une méthode `ficheTechnique()` qui présente l'instrument et toutes ses caractéristiques (incluant celles héritées de classes mères!)

Réfléchissons à la classe `Magasin`. Le magasin contient des stocks de chaque instrument, paie un loyer chaque mois de `montantLoyer` euros. Le magasin possède également une trésorerie (un `int`).

Écrivez la méthode `fournir(Instrument instrument, int quantite)` qui permet de remplir les stocks d'un instrument donné. Vous devrez vérifier que le magasin détient l'argent pour ses achats.

Écrivez une méthode `vendre(Instrument instrument)`, appelée lorsqu'un instrument est vendu. Les stocks et la trésorerie du magasin doivent évoluer en conséquence.

Si vous avez le temps...

Écrivez une fonction `simulerFonctionnement()` qui détaille le fonctionnement du magasin pour chaque mois ; un nombre (tiré au hasard) de clients visite le magasin chaque mois, et chaque client à une probabilité fixée par vos soins d'acheter un instrument : plus l'instrument est cher, plus la probabilité est faible !

Le magasin est-il rentable ? Sous quels paramètres ?