

Titre :
(LgP6Linf.EPS)
Auteur :
Adobe Illustrator(TM) 3.2
Aperçu :
Cette image EPS n'a pas été enregistrée
avec un aperçu intégré.
Commentaires :

Contrôle continu écrit de novembre texte et corrigé

Durée : **1h45** - Tous les documents *personnels*, manuscrits ou imprimés, sont autorisés.
Tous les appareils électroniques sont interdits et doivent être rangés.

NE RIEN ÉCRIRE AU CRAYON NI À L'ENCRE ROUGE S.V.P.

Ce sujet comporte 4 pages.

A étude d'un programme

On considère les deux classes suivantes :

```
1 class eleve {
2     private String nom;
3     private int niveau;
4     public int nbEleve;
5
6     // constructeurs
7     eleve(String n, int niv) {
8         super();
9         nom= n; niveau= niv;
10        nbEleve++;
11    }
12    eleve() { this("???", 0); }
13    String eleve(int niv) {
14        niveau= niv;
15        return nom;
16    }
17
18    // encapsulation
19    public String getNom() { return nom; }
20    public void SetNiveau(int niv) { niveau= niv; }
21
22    // redefinition de toString
23    String toString() {
24        return "Eleve "+nbEleve+" "+nom+" niveau "+niveau;
25    }
26 }

1 class Test {
2     public static void main(String a) {
3         Eleve e1= new Eleve("Paul",5);
4         System.out.println(e1.toString());
5         Eleve e2,e3=e1;
6         e1= new Eleve();
7         e2= new Eleve("Marie",2);
8         System.out.println(e1.toString()+"\n"+e2.toString()+"\n"+e3);
9     }
10 }
```

La classe `eleve` compile sans aucun message d'erreur. Elle présente pourtant plusieurs problèmes que nous résoudrons progressivement en cheminant parmi les questions qui suivent.

a1 ligne 1

Une convention d'écriture de Java n'a pas été respectée. Donner la nouvelle ligne. En conséquence, quelles autres lignes faudra-t-il modifier dans la suite du programme ?

a2 lignes 2 à 4

On a affaire à des définitions de variables de classe ou d'objets (attributs de classe ou d'instance). Pour chacune d'elles indiquez de quelle catégorie il s'agit ? Justifiez votre réponse.

a3 ligne 8

Quel est le principe fondamental qui permet d'écrire cette instruction (totalement valide).

a4 ligne 10

Quel est l'effet de cette instruction ?

a5 ligne 12

Que fait ce constructeur ?

a6 lignes 13 à 16

Quel est l'objectif de ces quatre lignes ? Rectifiez ce qui ne convient pas du tout : donnez la nouvelle écriture de ces lignes.

a7 ligne 22

Que signifie ce commentaire ?

a8 ligne 23

Dans quelle mesure cette ligne ne correspond-elle pas au commentaire ? Rectifiez.

Si vous avez correctement rectifié la classe `eleve`, la classe `Test` compile normalement. Elle présente pourtant un problème.

a9 ligne 2

Lorsqu'on lance l'exécution du programme en indiquant la classe `Test`, l'exécution ne démarre pas, et il est indiqué qu'on ne trouve pas la méthode `main`. Comment faut-il rectifier la ligne 2 pour pouvoir exécuter le programme ?

a10 ligne 5

Que veut dire `Eleve e3=e1` ; ? Attention : réfléchissez bien avant de répondre, car la réponse n'est pas forcément si simple qu'il y paraît.

a11 exécution de la ligne 8

Lorsqu'on lance l'exécution du programme on obtient l'affichage à l'écran suivant :

```
Eleve 1   Paul niveau 5
Eleve 1   ??? niveau 0
Eleve 1   Marie niveau 2
Eleve 1   Paul niveau 5
```

C'est la ligne 12 qui provoque l'affichage des trois dernières lignes. Expliquez l'affichage obtenu, et plus particulièrement la dernière ligne.

a12 résolution du problème à l'exécution : rectification de la déclaration de la classe `eleve`.

Le numéro de l'élève est toujours 1. Pourtant dans le constructeur de `eleve` on prend bien la précaution d'augmenter de 1 la variable `nbEleve`. Rectifier la déclaration de la variable `nbEleve` pour que l'on obtienne bien l'affichage à l'écran suivant :

```
Eleve 1   Paul niveau 5
Eleve 2   ??? niveau 0
Eleve 3   Marie niveau 2
Eleve 1   Paul niveau 5
```

Expliquez pourquoi on obtenait toujours `Eleve 1` ?

B Héritage et écriture de classes

On considère un établissement de restauration qui détermine des menus.

Un menu est composé de trois plats. Le plat dénommé principal, est à choisir parmi la liste suivante : bœuf, mouton, œuf, blé complet. Les deux autres plats, dénommés entrée ou dessert, sont à choisir parmi la liste suivante : salade, gâteau, friand, glace.

Un plat principal est donc un plat que l'on choisit dans la liste adéquate.

Un plat entrée-dessert est donc un plat que l'on choisit dans la liste adéquate.

Ces deux propositions montrent qu'il peut y avoir de l'héritage à partir d'une seule idée plat, et qu'un menu est constitué de plats.

Exemple de menu : `friand boeuf gateau`

Exemple d'un ensemble de menus :

```
glace oeuf glace
glace oeuf salade
friand boeuf gateau
salade blé complet salade
gateau mouton glace
```

Il faut définir les classes suivantes : une classe mère `UnPlat` et ses deux classes filles `Principal` et `EntreeDessert`, une classe `UnMenu` qui comporte les trois champs correspondant aux trois plats et la classe `Menus` pour l'ensemble des menus.

Très important Toute indentation mal faite ou incorrecte sera notée négativement.

b1 classe `UnPlat`

Elle définit un champ héritable, `plat`, qui est un nom. Elle définit également une méthode `autreChoix`, sans paramètre et qui fournit une chaîne en résultat, et dont les instructions ne peuvent être connues à ce niveau de la hiérarchie.

b2 classe `Principal`

Elle hérite de `UnPlat`. Elle comporte la liste des plats principaux, sous forme d'un tableau de chaînes; ce tableau est initialisé au sein de la définition de la classe. Elle comporte un champ `choisi` dont la valeur correspond à l'indice du plat choisi dans le tableau des plats principaux.

Le constructeur par défaut détermine le plat, en choisissant aléatoirement une des quatre possibilités. La méthode `toString` retourne en résultat le nom du plat choisi.

Les instructions de la méthode `autreChoix` seront définies à ce niveau de la hiérarchie. Son principe est simple : elle renvoie la liste des plats principaux, sauf celui qui a été choisi.

Donnez la définition de la classe `Principal`.

b3 classe `EntreeDessert`

Elle a grosso modo la même définition que la classe `Principal`, sauf la spécificité des plats.

Que faut-il modifier à la classe `Principal` pour obtenir une bonne définition de la classe `EntreeDessert` ?

b4 classe `UnMenu`

Pour décrire un menu, on a besoin de trois plats : `plat1`, `plat2` et `plat3`. Les deux extrêmes sont des entrées ou des desserts, et celui du milieu un plat principal.

C'est le constructeur par défaut qui crée les trois objets correspondant aux trois plats. La méthode `toString` renvoie en résultat, sur une même ligne, le nom des trois plats, puis passe à la ligne.

Définir la classe `UnMenu`.

b5 classe Menus

Le nombre de menus à définir est fixé par une « constante » placée dans la définition de la classe Menus. L'ensemble des menus sera stocké dans un tableau de UnMenu.

Le constructeur par défaut crée les menus, et la méthode `toString` liste les menus, un par ligne.

Définir la classe Menus.

b6 classe de test

Donner la définition d'une classe permettant de tester le fonctionnement de ce programme.

b8 Autre héritage

Les classes Principal et EntreeDessert ont beaucoup de choses en commun : elles ne diffèrent en fait que par la liste des plats.

Que proposez-vous pour éviter d'avoir à dupliquer toutes ces instructions ? Pensez à une forme particulière d'héritage et faite attention à la restriction liée à `extends`.