le cnam

Année universitaire 2015-2016

SUJET NFA032 : programmation java : programmation objet

Examen 1ère session du 11/02/2016

Enseignant responsable : François Barthélemy

Durée : 3 heures.

consignes

Calculatrice non autorisée

Tous documents autorisés

Les téléphones mobiles et autres équipements communicants (exemple : PC, tablette, etc) doivent être éteints et rangés dans les sacs pendant toute la durée de l'épreuve.

Sujet de 5 pages y compris celle-ci.

Vérifiez que vous disposez bien de la totalité des pages du sujet en début d'épreuve et signalez tout problème de reprographie le cas échéant.

Exercice 1 : programmation objet (5 points)

Question 1

Ecrivez une classe permettant de représenter un plat proposé par un restaurant. Ce plat a un intitulé, un prix et une catégorie parmi : entrée, plat principal et dessert. Cette classe comportera un constructeur et une méthode d'affichage.

Question 2

Ecrivez une classe représentant un repas dans le restaurant, comportant une entrée, un plat, un dessert. Cette classe devra avoir un constructeur qui initialise les variables, une méthode d'affichage et une méthode qui calcule le prix total du repas qui est la somme des prix des trois plats. La classe devra permettre de vérifier, soit par le constructeur, soit par une méthode qu'il y a bien exactement un plat de chaque catégorie dans le repas.

Question 3

On veut maintenant changer la classe des repas pour que l'entrée et le dessert soient facultatif. Le repas comprend au plus une entrée, mais éventuellement il n'en a aucune. Idem pour le dessert. Le plat principal reste obligatoire. Donnez le code de la classe modifiée (ou seulement les différences si elles sont peu nombreuses).

Exercice 2 : références (3 points)

```
class Grand{
    Petit[] tab;
    Grand(int x){
        tab = new Petit[1];
        tab[0] = new Petit(x);
    }
}
class Petit{
    int x;
    Petit(int xx){
        x = xx;
    }
}
public class Ref16{
    public static void main(String[] args){
        Grand[] chose;
        chose = new Grand[1];
        chose[0] = new Grand(1);
    }
}
```

Question 1

Représentez au moyen d'un petit dessin les objets et tableaux existant à la fin de l'exécution du programme donné ci-dessus. Chaque objet et chaque tableau sera représenté par un rectangle et chaque référence par une flèche ou une adresse. Attention : on ne vous demande pas un diagramme de classe. C'est chaque objet créé, chaque tableau et chaque référence qui doit être représenté.

Question 2

Donnez le code java permettant d'afficher à l'écran la valeur de la variable `x` de chaque objet instance de `Petit` créé par ce programme.

Exercice 3 : exceptions (2,5 points)

```
public class Excep4 {
   public static void m3(int x, int y){
      if (x==1)
           throw new Err1();
      else if (x>y) {
           throw new Err2();
      }
      Terminal.ecrireStringln("Fin␣normale␣p3");
   }
   public static void m2(int x, int y){
      try {
              m3(x, y);
              Terminal.ecrireStringln("Fin␣normale␣p2");
           } catch (Err2 e) {
              Terminal.ecrireStringln("Recuperation␣p2");
      }
   }
   public static void m1(int x, int y){
     try {
         m2(x,y);
         Terminal.ecrireStringln("Fin␣normale␣p1");
     } catch (Err1 e ){
         Terminal.ecrireStringln("Recuperation␣p1");
     }
   }

   public static void main(String[] args) {
     int a = Terminal.lireInt();
     int b = Terminal.lireInt();
     m1(a,b);
     Terminal.ecrireStringln("Fin␣programme␣");
   }
}
class Err1 extends Error{}
class Err2 extends Error{}
```

Donnez les messages affichés par l'exécution de ce programme si les valeurs lues pour les variables a et b sont respectivement :

1. a=0 et b=5
2. a=1 et b=1
3. a=2 et b=1

Exercice 4 : interface/héritage (5 points)

Les carnivores mangent les herbivores et les herbivores meurent quand ils sont mangés. Les carnivores et les herbivores sont des animaux. Les animaux sont vivants à leur création et peuvent mourir. Ils mangent de la nourriture. Les légumes sont de la nourriture.

1. créez des classes pour représenter les différentes réalités décrites ci-dessus. Mettez dans ces classes le minimum de choses, mais suffisament pour représenter les notions évoquées dans le texte.
2. donnez le code correspondant à un loup qui mange un lapin. Ne créez pas de classe spéciale pour les loups et les lapins : utilisez les classes et méthodes écrites à la question précédente.
3. créez une sous-classe des légumes qui représente les légumes empoisonnés. Quand un animal mange une nourriture empoisonné, il meure. De plus il est lui-même une nourriture empoisonnée pour les carnivores.
4. distinguez parmi les carnivores les charognards qui ne mangent que les animaux déjà morts et les carnassiers qui ne mangent que les animaux vivants qu'ils tuent eux-mêmes. Pour cela vous écrirez deux classes.

Exercice 5 : structures récursives (5 points)

On donne la classe récursive suivante qui sert à représenter un élément d'une liste de course. On la représente au moyen d'objet instances de la classe `UnArticle` : chaque objet représente un article à acheter et contient aussi une référence à l'article suivant.

```
class UnArticle{
    String nom,unite; // unite: kg, litre, piece, douzaine
    double quantite;
    int type;// 0: fruit, 1: légumes, 2: viande
    UnArticle articleSuivant;
    UnArticle(String n, double q, String u, int t, UnArticle ua){
        nom = n;
        unite = u;
        quantite = q;
        type=t;
        articleSuivant=ua;
    }
}
```

– écrivez une méthode d'affichage à ajouter à la classe donnée.

– mettez dans une variable une liste de course contenant 1,5 kg d'oranges, une salade et 3 steaks hachés.
– écrivez l'appel de méthode permettant d'afficher cette liste et dîtes dans quel ordre les éléments s'affichent.
– on péfère les listes où tous les éléments d'un même type sont contigus. Cela évite de se tromper quand on est chez un marchand d'un type donné (par exemple le boucher qui ne vend que de la viande). Ecrivez une méthode qui vérifie si une liste respecte cette propriété ou pas (résultat de type boolean).