Année universitaire 2015 – 2016

Sujet UE NFA032: Programmation avec Java : programmation objet

Examen deuxième session : 21/04/2016

Responsable : F. Barthélemy

Durée : 3 heures

Consignes

Aucun document n'est autorisé.

Calculatrice non autorisée

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

Sujet de 5 pages, celle-ci comprise.

Le barème est donné à titre purement indicatif. Il est susceptible de changer

→ *Vérifiez que vous disposez 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 une entrée de dictionnaire et qui comporte les informations suivantes : le mot ; sa catégorie grammaticale (nom, verbe, adjectif, etc) ; sa définition. Cette classe comportera un constructeur et une méthode d'affichage.

Question 2

On veut maintenant représenter des *formes fléchies* des mots, c'est-à-dire des formes conjuguées des verbes et des formes accordées des noms et adjectifs. Une forme fléchie comprendra une entrée de dictionnaire (un objet de la classe définie à la question 1) et des informations de flexion (temps, personne, genre, nombre, etc). Cette classe comportera un constructeur et une méthode d'affichage.

Question 3

Donnez une méthode main qui crée et affiche des objets des deux classes écrites précédemment.

Exercice 2 : références (3 points)

```
class Grand{
    int[] tab;
    Grand(int x){
        tab = new int[x];
        for (int i=0; i<x; i++){
            tab[i]=x+i;
        }
    }
}
public class Ref17{
    public static void main(String[] args){
        Grand[] chose;
        chose = new Grand[1];
        chose[0] = new Grand(2);
    }
}
```

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 tous les entiers contenus dans des tableaux. Pour cette question, il n'est pas permis de changer le code de la classe Grand.

Exercice 3 : exceptions (3,5 points)

```
public class Excep8 {
    public static void m2(int x, int y) throws Err1, Err2{
        if (x==1)
            throw new Err1 ();
        else if (x>y) {
            throw new Err2 ();
        }
        Terminal.ecrireStringln("Fin_normale_m2");
    }
    public static void m1(int x, int y) throws Err1, Err2{
        try {
            m2(x,y);
            Terminal.ecrireStringln("Fin_normale_m1");
        } catch (Err1 e ){
            Terminal.ecrireStringln("Recuperation_Err1");
        } catch (Err2 e ){
            m2(x-1,y);
        }
    }

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

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

1. a=1 et b=2
2. a=2 et b=1
3. a=2 et b=2
4. a=3 et b=2
5. a=4 et b=2

Exercice 4 : interface/héritage (4,5 points)

```
interface Vegetal{
    void estIngéré();
}
interface Herbivore{
    void mange(Vegetal v);
    void meurt();
}
class Animal{
    boolean vivant = true;
    int poids;
    Animal(int p){
        poids=p;
    }
    public void meurt(){
        vivant = false;
    }
}
class Mouton extends Animal implements Herbivore{
    Mouton(int p){
        super(p);
    }
    public void mange(Vegetal v){
        v.estIngéré();
    }
    void bele(){
        System.out.println("bêêêêê");
    }
}
```

Question 1

Lesquelles des lignes suivantes sont correctes ?

1. Vegetal veg = new Vegetal();
2. Vegetal veg = new Mouton();
3. Animal an = new Animal();
4. Animal an = new Mouton();
5. Animal an = new Herbivore();
6. Mouton mou = new Animal();
7. Herbivore herb = new Animal();
8. Herbivore herb = new Mouton();

Question 2

```
Animal andre = new Animal();
```

```
Animal paul = new Mouton();
Herbivore sylvie = new Mouton();
```

Donnez la liste des variables et des méthodes de chacun des trois objets créés.

Question 3

Avec les mêmes déclarations qu'à la question précédente, donnez la liste des méthodes que l'on peut appeler sur les variables andre, paul et sylvie.

Exercice 5 : structures récursives (4,5 points)

On donne la classe récursive suivante qui sert à représenter un élément d'une liste de course. On la représente au moyen d'objets 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;
    UnArticle articleSuivant;
    UnArticle(String n, double q, String u, UnArticle ua){
        nom = n;
        unite = u;
        quantite = q;
        articleSuivant=ua;
    }
    public static void main(String[] args){
        UnArticle una = new UnArticle("oranges",3.0,"kg",null);
        una = new UnArticle("lait",0.75,"litre",una);
        una = new UnArticle("steaks",2.0,"pièce",una);
    }
    public static void main2(String[] args){
        UnArticle una1 = new UnArticle("oranges",3.0,"kg",null);
        UnArticle una2 = new UnArticle("lait",0.75,"litre",una1);
        UnArticle una3 = new UnArticle("steaks",2.0,"pièce",una1);
    }
}
```

1. écrivez une méthode d'affichage à ajouter à la classe `UnArticle`.
2. dessinez la structure `una` construite dans la méthode `main`.
3. le code de la méthode `main2` est-il correct? Si oui, donnez un dessin illustrant le contenu des variables `una1`, `una2` et `una3` en fin d'exécution de la méthode.