

NFA032 Programmation objet avec Java

CNAM-centre d'enseignement de Paris

Première session 2015 semestre 1

Documents et calculatrice interdits. Le barème est donné à titre indicatif.

Exercice 1 : programmation objet (5 points)

Question 1

Ecrivez une classe représentant une équipe de football avec un nom, un sigle et une ville de rattachement, un constructeur et une méthode d'affichage.

Question 2

Ecrivez une classe représentant un match de championnat avec une équipe qui reçoit à domicile, l'autre équipe qui est en déplacement, un statut qui peut être *à jouer*, *en cours* ou *terminé* et un score où l'on a le nombre de but de chaque équipe. Ecrivez les méthodes permettant de faire évoluer le statut et le score au fil du temps. Ecrivez une méthode permettant d'obtenir le score qui lève une exception si le match n'est pas encore commencé.

Question 3

Ecrivez une méthode main avec d'abord la création d'un match programmé entre deux équipes réelles ou fictives, puis le changement de statut avec le début du match et un score en cours de match de 1-1 (1 but pour chaque équipe).

Exercice 2 : références (3 points)

```
class Grand{
    Petit p;
    Petit[] tab;
    Grand(int x, int y){
        tab = new Petit[3];
        p = new Petit(x);
        tab[0]=p;
        tab[1]=new Petit(y);
    }
}
```

```

class Petit{
    int x;
    Petit(int xx){
        x = xx;
    }
}
public class Ref15{
    public static void main(String[] args){
        Grand legrand;
        legrand = new Grand(5,7);
    }
}

```

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 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 Excep6 {
    public static void meth3(int x){
        if (x==0)
            throw new PresquePas();
        else if (x>2) {
            throw new PasDuTout();
        }
        Terminal.afficheStringln("Fin_normale_meth3");
    }
    public static void meth2(String s){
        if (s==null){
            throw new PresquePas();
        }
        try {
            meth3(s.length());
            Terminal.afficheStringln("Fin_normale_meth2");
        } catch (PresquePas e) {
            Terminal.afficheStringln("PresquePas_dans_meth2");
        }
    }
}

```

```

public static void meth1(char c, String s){
    try {
        meth2(s);
        Terminal. ecrireStringln ("Fin_normale_meth1");
    } catch (PasDuTout e){
        Terminal. ecrireStringln ("PasDuTout_dans_meth1");
    }
}

public static void main(String[] args) {
    try{
        meth1('a', null);
        Terminal. ecrireStringln ("Fin_normale_main_premier_try");
    } catch (PasDuTout e){
        Terminal. ecrireStringln ("PasDuTout_dans_main");
    } catch (PresquePas e){
        Terminal. ecrireStringln ("PresquePas_dans_main");
    }
    try{
        meth1('z', "oui");
        Terminal. ecrireStringln ("Fin_normale_main_second_try");
    } catch (PasDuTout e){
        Terminal. ecrireStringln ("PasDuTout_dans_main");
    } catch (PresquePas e){
        Terminal. ecrireStringln ("PresquePas_dans_main");
    }
    Terminal. ecrireStringln ("Fin_normale_main");
}
}
class PasDuTout extends Error{}
class PresquePas extends Error{}

```

Donnez les messages affichés par l'exécution de ce programme.

Exercice 4 : structures récursives (5 points)

On veut représenter la tournée d'un facteur comme une suite de boîtes aux lettres ordonnée. On suppose que la tournée est dans une seule ville, ce qui permet de simplifier la représentation des adresses : chaque adresse est un numéro de rue et un nom de rue. Par ailleurs, à certains numéros, par exemple pour un immeuble, il y a plusieurs boîtes à la même adresse : on les distingue par le nom du titulaire de la boîte à lettres.

1. écrivez une classe récursive avec les informations permettant de représenter une boîte et une variable contenant une référence vers la boîte suivante.
2. écrivez une méthode permettant d'afficher une tournée de facteur ainsi représentée.
3. écrivez une méthode **récursive** qui teste si une boîte donnée appartient à la tournée d'un facteur. Cette boîte sera identifiée par son adresse et le nom du titulaire.
4. écrivez une méthode qui ajoute une boîte dans une tournée existante. Pour déterminer la position de cette nouvelle boîte on donnera le nom et l'adresse de la boîte qui la précède dans la tournée.

Par exemple, on pourra demander à ajouter la boîte Martin au 5 rue du Bois après la boîte Durand au 3 rue du Bois.

Exercice 5 : interface/héritage (5 points)

On veut représenter en programmation objet des expressions arithmétiques sur les nombres entiers. Une expression arithmétique peut être soit une valeur littérale (un nombre) soit un opérateur appliqué à deux expressions arithmétiques. Une expression peut être calculée pour donner un résultat qui est un entier.

Question 1

Dans une version simplifiée, un opérateur ne peut être appliqué qu'à deux nombres. Ecrivez les classes et interfaces nécessaires pour pouvoir représenter une valeur littérale, les expressions avec opérateurs + et les expressions avec opérateur * (on pourrait faire aussi les autres opérateurs de la même façon).

Ecrivez une méthode main qui utilise ces classes. Déclarez dans le main une variable `var` dans laquelle sont affectées successivement une valeur littérale et une expression arithmétique avec opérateur. Puis le programme affiche le résultat du calcul pour cette expression avec opérateur. Un squelette de cette méthode main est proposé :

```
public static void main(String[] args) {
    XXX var;
    var = .... donner le code pour créer un objet représentant
               le nombre 17 ...
    var = .... donner le code pour créer un objet représentant
               l'expression 5 * 3 .....
    Terminal.ecrireIntln(... donner le code pour calculer la valeur
                          de l'expression contenue dans var ....);
}
```

Question 2

Modifiez les classes des opérateurs pour permettre que les deux opérandes puissent être aussi bien d'autres expressions avec opérateur que des valeurs littérales. Par exemple, on veut pouvoir représenter $5*3$ (comme à la question précédente) mais aussi $5*(3+2)$ et $(5+(2*3))+(3*2)$.