

# Programmation orientée objet : Java - TP 7 -

Romain Péchoux

23/11/2009

## Exercice 1 : Les packages

- Créer une nouvelle classe. Importer la classe Scanner du package java.util et utiliser la afin d'écrire un programme qui termine si le mot saisi est "adios amigo" et qui demande à l'utilisateur de saisir du texte dans tous les autres cas. Comment peut-on importer toutes les classes du package java.util ? De quelles manières différentes peut-on faire référence aux membres de la classe Scanner ?
- Créer une classe Entier contenant un attribut entier x sans spécification d'accès, un constructeur sans argument initialisant l'entier à 0. Mettre cette classe dans un package "nombres", puis déplacer le fichier dans le sous-répertoire correspondant. Ecrire une classe exécutable, créer une instance r de la classe Entier et afficher la valeur r.x. Importer le package nombres. Que se passe-t-il ? Résoudre ce problème. Que se passe-t-il si l'on utilise le nom complet nombres.Entier au lieu de la commande import ?

## Exercice 2 : Comprendre les exceptions

Soient les deux classes d'exception E1 et E2 définies pour j égal à 1 (respectivement 2) par :

```
public class Excj extends Exception {  
    public Excj(String s) {  
        super(s);  
    }  
}
```

On considère le programme suivant :

```
public class ProgrammeException {  
    private int x;  
    public ProgrammeException (int x) {  
        this.x = x;  
    }  
    public int getX() {
```

```

        return x;
    }
    public void methode1(int y) throws Exc1,Exc2 {
        if (y == 0) throw new Exc1("division par zero");
        if (x * y < 0) throw new Exc2("parametre de signe oppose"); x = x / y ;
    }
    public void methode2(int y) throws Exc1 {
        try { methode1(y);
        }catch (Exc2 excpt) {
            System.out.println(excpt.getMessage());
        }
    }
    public static void main(String[] args) throws Exc1 {
        ProgrammeException pe = new ProgrammeException(Integer.parseInt(args[0]));
        pe.methode2(Integer.parseInt(args[1]));
        System.out.println("x = " + pe.getX());
    }
}

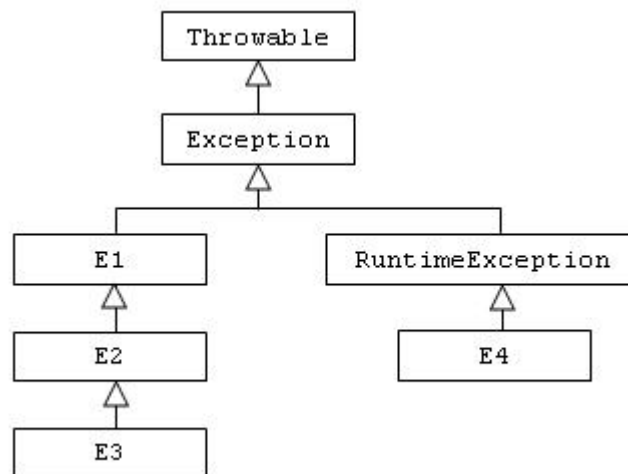
```

Qu'affiche le programme pour les lignes de commandes suivantes (à résoudre sur papier avant de tester sur machine) :

- java ProgrammeException 30 5
- java ProgrammeException 20 -5
- java ProgrammeException 113 0

### Exercice 3 :

On considère la hiérarchie de classe suivante :



Soit une classe Classe1 possédant deux méthodes publiques :

- `public void methode1()`, dont le corps peut lever des exceptions de type E1, E2, E3 ou E4,
- `public void methode2()` ne levant pas d'exceptions.

Soit une classe `Classe2` définissant une méthode `public void methode3(Classe1 x)` dont le corps consiste à envoyer un message `methode1` puis un message `methode2` à l'objet `x` passé en paramètre. Au cas où l'appel de `methode1` leverait une exception, les traitements suivants sont réalisés :

- Si l'exception est de type E2, le message contenu dans l'objet exception est affiché sur la console,
- Si l'exception est de type E3, une trace de la pile des appels de méthodes ayant conduit à la levée de l'exception est affichée sur la console,
- Si l'exception est de type E1 ou E4, elle est simplement retransmise vers le bloc appelant.
- Dans tous les cas de figure, levée ou non d'une exception par `m1`, le message `m2` devra être envoyé à l'objet `x` avant le retour de la méthode `methode1`.

Ecrire le code java du programme en respectant les spécifications ci-dessus.

Ajouter à la classe `Classe2` une seconde méthode `public void methode4(ClasseX x)` dont le corps consiste à envoyer un message `m1` puis un message `m2` à l'objet `x` passé en paramètre. Au cas où l'appel de `m1` leverait une exception, les traitements suivants sont réalisés :

- Si l'exception est de type E1, le message contenu dans l'objet exception est affiché sur la console,
- Si l'exception est de type E2, une trace de la pile des appels de méthodes ayant conduit à la levée de l'exception est affichée sur la console,
- Si l'exception est de type E3 ou E4, elle est simplement retransmise vers le bloc appelant.
- Dans tous les cas de figure, levée ou non d'une exception par `m1`, le message `m2` devra être envoyé à l'objet `x` avant le retour de la méthode `methode2`.

Ecrire le code java du programme en respectant les spécifications ci-dessus.

## Exercice 4 : Comptes bancaires et leurs exceptions

Reprendre la classe des Comptes Bancaires du TD4. Modifier la afin de traiter les exceptions (solde négatifs, saisie d'une chaîne de caractères au lieu d'un entier, ...).