

Travaux pratiques d'informatique N° 16

Le but de cette séance est de vous permettre de vérifier vos connaissances concernant le mécanisme d'exceptions Java.

1. Créer un nouveau projet Eclipse appelé **PrTP16Exo1**, en suivant la démarche suivante :
 - a) depuis le bureau virtuel du CMS, copier les fichiers **CMSSStudent.java** et **CP_TP16Exo1.java** dans le package **cms_tp16** du projet ;
 - b) comprendre le projet et anticiper les résultats affichés suite à son exécution ;
 - c) exécuter le projet et comparer les résultats obtenus avec ceux anticipés ;
 - d) en utilisant le mécanisme d'exceptions, modifier la méthode **setName()** de la manière suivante :
 - i. tout nom commençant par une majuscule sera accepté comme valide ;
 - ii. tout nom commençant par une minuscule sera modifié en changeant sa première lettre en majuscule ;
 - iii. un nom ne commençant ni par une majuscule ni par une minuscule produira l'arrêt intempestif du programme, après l'affichage d'un message d'erreur;
 - e) en utilisant le mécanisme d'exceptions, modifier la méthode **setAge()** de la manière suivante :
 - i. tout age compris entre 16 et 23 ans sera accepté comme valide ;
 - ii. pour tout age strictement inférieur à 16 ans, la valeur limite inférieure de 16 ans sera retenue ;
 - iii. pour tout age strictement supérieur à 23 ans, la valeur limite supérieure de 23 ans sera retenue ;
 - f) exécuter le projet et analyser les résultats obtenus.

Indications pour les solutions faisant appel au mécanisme d'exceptions :

- il faut ajouter au projet une source qui définit une nouvelle classe **NameException**, descendante directe de la classe **java.lang.Exception** ;
- il faut ajouter au projet une source qui définit une nouvelle classe **AgeException**, descendante directe de la classe **java.lang.RuntimeException** ;
- le constructeur de la classe **CMSStudent** avec 2 arguments doit traiter partiellement les exceptions de type **NameException**, à savoir le cas **d) ii.**, et déléguer le traitement du cas **d) iii.** à la méthode **main()** ;
- les exceptions de type **AgeException** doivent être traitées, dans une première variante, directement par la **JVM** et, dans une deuxième variante, par la méthode **main()** ;
- il convient d'employer également des blocs **finally**, afin de mieux comprendre leurs avantages et leur fonctionnement ;
- il est recommandé de modifier les noms et les âges utilisés dans la méthode **main()** pour tester le traitement des exceptions.

2. Soit un fichier source **CP_TP16Exo2.java** avec le contenu ci-dessous :

```
package cms_tp16;
class UneClasse {
    public char uneMethode(char arg) throws MajException {
        try {
            if(arg >= 'a' && arg <= 'z')        throw new MinException();
            else if(arg >= 'A' && arg <= 'Z')    throw new MajException();
            else if(arg >= '0' && arg <= '9')    throw new ChiffreException();      }
        catch(MinException refObjMinExp) {
            System.out.println("\nJe suis le gestionnaire de MinException !");
            return arg;                                }
        finally { System.out.println("\nJe suis le bloc finally pour uneMethode !");      }

        System.out.println("\nAprès le bloc finally pour uneMethode !");
        return arg;
    } //fin de la méthode uneMethode
} //fin de la classe UneClasse
```

```

class MinException extends Exception { }

class MajException extends Exception { }

class ChiffreException extends RuntimeException { }

public class CP_TP16Exo2 {
    public static void main(String args[ ]) {
        UneClasse refObj = new UneClasse();
        char tab[ ] = {'?', '?', '?', '?'};

        try{
            tab[0] = refObj.uneMethode('b');    //ligne 1
            tab[1] = refObj.uneMethode('H');    //ligne 2
            tab[2] = refObj.uneMethode('7');    //ligne 3
            tab[3] = refObj.uneMethode('@');    //ligne 4
        }
        catch(MajException refObjMajExp) {
            System.out.println("\nJe suis le gestionnaire de MajException !");
        }
        catch(ChiffreException refObjChiffreExp) {
            System.out.println("\nJe suis le gestionnaire de ChiffreException !");
        }

        System.out.println();
        for(int i=0; i<4; i++)      System.out.print(tab[i]);
        System.out.println( );

        System.out.println( "\nFin de la methode main!" );
    }
} //fin de la classe CP_TP16Exo2

```

Quels sont les résultats affichés suite à l'exécution d'un projet ayant comme source le fichier présenté ci-dessus ?

En tenant compte de toutes les combinaisons possibles, modifier les quatre lignes du bloc **try** de la méthode **main()** des trois façons suivantes :

- en commentant trois lignes ;
- en commentant deux lignes ;
- en commentant une ligne.

Quels seront les nouveaux résultats affichés après chaque modification ?