

Vendredi 6 février 2019

Q1)

```
Exn.java
1 class Essai1Exception extends Exception{ //ss-classe de la classe Exception
2     Essai1Exception(String s){
3         super(s);
4     }
5 }
6 class Essai2Exception extends Essai1Exception{ //superclasse directe de Essai1Ex..
7     Essai2Exception(String s){
8         super(s);
9     }
10 }
11 //classe principale d'exception Exn
12 public class Exn {
13     static void throwEssais(int i) throws Exception {
14         switch (i){ //structure switch case
15             case 1: System.out.println("Lancement de Essai1Exception") ;//cas n°1
16                 throw new Essai1Exception ("Essai1Exception de throwEssais") ;
17             case 2: System.out.println("Lancement de Essai2Exception") ; //cas n°2
18                 throw new Essai2Exception ("Essai2Exception de throwEssais") ;
19             default: System.out.println("Lancement de Exception") ;//cas par défaut-càd pour tous les autres cas
20                 throw new Exception("Exception de throwEssais") ; //msg retourné
21         }
22     }
23     //méthode principale
24     public static void main(String[] args) {
25         for (int i = 1 ; i <=3 ; i++){ //i variant de 1 à 3
26             try { //appel d'instructions pouvant causer des exceptions
27                 throwEssais(i) ; //fait appel à throwEssais (switch-case)
28                 System.out.println("Retour d'exception") ;//cas inaffiché car c exécuté après throwEssais
29                 System.out.print("test"); //idem
30             }
31             catch (Essai2Exception e){
32                 System.out.println("Catch Essai2: " + e.getMessage()) ;//gestion de l'Essai2Exception
33             }
34             catch (Essai1Exception e){
35                 System.out.println("Catch Essai1: " + e.getMessage()) ;//gestion de l'Essai1Exception
36             }
37             catch (Exception e){ System.out.println("Catch Exception : " + e.getMessage()) ;//case par défaut
38             }
39             finally { System.out.println("Finally de main.") ; } //bloc traité à chq fin d'exécution
40         }
41     }
42 }
```

Code :

La Classe Exception : est la superclasse directe /classe parent de l'Essai1Exception

La Classe l'Essai2Exception: est la superclasse directe /classe parent de l'Essai2Exception

La variable s de la superclasse est réutilisé grâce au super, il semblerait qu'elle permet de retourner la chaîne de caractères

On définit plusieurs classes qui s'héritent les unes des autres

Puis on utilise une structure switch-case jusqu'à 2 cas et le cas par défaut,

On définit la méthode principale : on fait varier i de 1 à 3 grâce à la boucle for puis on essaye de faire appel à throwEssais

Lors du try s'il a déjà appelé l'exception (ici throwEssais) alors seules les case sont exécutées on remarque que les autres ne sont pas du tout affichées.

Les catches Exceptions sont définis ils font appel aux messages des cases en fonction de la valeur de i  
D'abord au System.out.print du case throwsEssais  
e.getMessage permet de faire appel au message throw new Exception du case throwEssais  
Le finally est toujours exécuté à chaque try ou catch.

Le mot-clé throws ici pour throwsEssais lié avec Exception signifie que le programme pourra générer des exceptions par conséquent le switch-case contient les exceptions

Exception est déjà utilisée par défaut dans java.util

L'expression throw new permet de retourner les Messages d'exception de la classe et sera intercepté par un catch

Les autres parties ont été commentées

De plus si i n'est pas défini alors le cas par défaut s'affiche ex : for (int i = 0 ; i <= 3 ; i++)

### **Résultat :**

L'Essai1Exception est exécuté puis l'Essai2Exception (car i vaut 2) puis l'Exception (i vaut 3 cas par défaut). Le programme s'arrête.

### **Résultat code original :**

```
Lancement de Essai1Exception  
Catch Essai1: Essai1Exception de throwEssais  
Finally de main.  
Lancement de Essai2Exception  
Catch Essai2: Essai2Exception de throwEssais  
Finally de main.  
Lancement de Exception  
Catch Exception : Exception de throwEssais  
Finally de main.
```

Q2)

Code :

```
*Exnbis.java
1 class Essai1Exception extends Exception{ //ss-classe de la classe Exception
2     Essai1Exception(String s){
3         super(s) ;
4     }
5 }
6 class Essai2Exception extends Exception{ //superclassedirecte de Exception
7     Essai2Exception(String s){
8         super(s) ;
9     }
10 }
11 //classe pricipale d'exception Exnbis
12 public class Exnbis {
13     static void throwEssais(int i) throws Exception {
14         switch (i){ //structure switch case
15             case 1: System.out.println("Lancement de Essai1Exception") ;//cas n°1
16                 throw new Essai1Exception ("Essai1Exception de throwEssais") ;
17             case 2: System.out.println("Lancement de Essai2Exception") ; //cas n°2
18                 throw new Essai2Exception ("Essai2Exception de throwEssais") ;
19             default: System.out.println("Lancement de Exception") ;//cas par défaut=càd pour tous les autres cas
20                 throw new Exception("Exception de throwEssais") ; //lmsg retourné
21         }
22     }
23     //méthode principale
24     public static void main(String[] args) {
25         for (int i = 1 ; i <=3 ; i++){ //i variant de 1 à 3
26             try { //appel d'instructions pouvant causer des exceptions
27                 throwEssais(i) ; //fait appel à throwEssais (switch-case)
28                 System.out.println("Retour d'exception") ;//cas inaffiché car c exécuté après throwEssais
29             }
30             catch (Essai1Exception e){
31                 System.out.println("Catch Essai1: " + e.getMessage()) ;//gestion de l'Essai1Exception
32             }
33             catch (Essai2Exception e){
34                 System.out.println("Catch Essai2: " + e.getMessage()) ;//gestion de l'Essai2Exception
35             }
36             catch (Exception e){ System.out.println("Catch Exception : " + e.getMessage()) ;//case par défaut
37             }
38             finally { System.out.println("Finally de main.") ;} //bloc traité à chq fin d'exécution
39         }
40     }
41 }
42 }
43 }
44 }
```

On constate que Essai2Exception est une super classe directe de Exception

On voit que l'ordre des catches dans la boucle for a changé, elle n'a pas d'influence sur l'exécution

Auparavant la classe Essai2Exception héritée de la classe Essai1Exception et de la classe Exception (car Essai1Exception hérite de cette classe Exception)

Alors que désormais Essai2Exception n'hérite plus de la classe Essai1Exception

Résultat :

Le résultat reste identique.

```
Lancement de Essai1Exception
Catch Essai1: Essai1Exception de throwEssais
Finally de main.
Lancement de Essai2Exception
Catch Essai2: Essai2Exception de throwEssais
Finally de main.
Lancement de Exception
Catch Exception : Exception de throwEssais
Finally de main.
```

### Test de ce programme avec de légère modification :

Si on met Essai2Exception extends Essai1Exception et que le catch est d'abord Essai2Exception une erreur est provoquée.

En effet cela ne marche pas car Essai1Exception doit retourner le message d'exception pour Essai2Exception

Par conséquent Essai2Exception extends Exception marche

Donc : on en conclut qu'extends Exception est pratique dans ce type de cas.

### Code :

```
class Essai1Exception extends Exception{
    Essai1Exception(String s){ super(s) ; }
}
class Essai2Exception extends Essai1Exception{
    Essai2Exception(String s){ super(s) ;}
}
//classe principale d'exception Exnbis
public class Exnbis {
    static void throwEssais(int i) throws Exception {
        switch (i){ //structure switch case
            case 1: System.out.println("Lancement de Essai1Exception") ;//cas n°1
                throw new Essai1Exception ("Essai1Exception de throwEssais") ;
            case 2: System.out.println("Lancement de Essai2Exception") ; //cas n°2
                throw new Essai2Exception ("Essai2Exception de throwEssais") ;
            default: System.out.println("Lancement de Exception") ;//cas par défaut=càd pour tous les autres cas
                throw new Exception("Exception de throwEssais") ; //Le msg d'exception retourné
        }
    }
    //méthode principale
    public static void main(String[] args) {
        for (int i = 1 ; i <=3 ; i++){ //i variant de 1 à 3
            try { //appel d'instructions pouvant causer des exceptions
                throwEssais(i) ; //fait appel à throwEssais (switch-case)
                System.out.println("Retour d'exception") ;//cas inaffiché car c'est exécuté après throwEssais
            }
            catch (Essai2Exception e){ System.out.println("Catch Essai2: " + e.getMessage()) ;} //gestion de l'Essai2Exception
            catch (Essai1Exception e){ System.out.println("Catch Essai1: " + e.getMessage()) ;} //gestion de l'Essai1Exception
            catch (Exception e){ System.out.println("Catch Exception : " + e.getMessage()) ;} //case par défaut
            finally { System.out.println("Finally de main.") ;} //bloc traité à chq fin d'exécution
        }
    }
}
```

### Résultat :

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    Unreachable catch block for Essai2Exception. It is already handled by the catch block for Essai1Exception
    at Exnbis.main(Exnbis.java:31)
```