

TP JAVA Les Exceptions

Exercice n°1 :

Soit le programme EssaiException.java suivant :

```
import java.util.Scanner;
public class EssaiException {
    public static void main(String[] args) {
        int a, b, res;
        Scanner clavier = new Scanner(System.in);
        a = clavier.nextInt(); b = clavier.nextInt();
        res = a / b;
        System.out.println("le résultat de " + a + " divisé par " + b + " est " + res);
        System.out.println ("Fin du programme") ;
    }
}
```

Si vous êtes dans un état normal, vous devez être persuadé que notre programme recèle une faille importante car nous divisons sans vérifier que le diviseur n'est pas nul. Néanmoins, lancez le programme en saisissant une valeur nulle pour b.

Q1 : Le programme s'est-il exécuté correctement ?

Q2 : Le message "Fin du programme" est-il apparu ?

Q3 : Quelle exception a été levée par la machine Java ?

Nous allons maintenant faire en sorte que le programme ne se termine pas aussi brutalement, et nous renseigne un peu plus. Pour cela, nous allons mettre en place un bloc try/catch afin d'attraper l'exception levée précédemment

```
try {
    res = a / b;
    System.out.println("le résultat de " + a + " divisé par " + b + " est " + res);
}
catch (ArithmeticException e) {
    System.out.println ("oop ! un problème dans la division ") ;
    System.out.println ("le message officiel est " + e.getMessage()) ;
}
System.out.println ("Fin du programme") ;
```

Relancez le programme en saisissant une valeur nulle pour b.

Q4 : Le programme a-t-il affiché qu'il y avait un problème dans la division ?

Q5 : Le message "Fin du programme" est-il apparu ?

Q6 : Quel est le message d'erreur officiel correspondant à une telle exception ?

A la suite de l'instruction catch vous allez maintenant rajouter le bloc finally suivant :

```
finally {  
    System.out.println("le bloc finally sera toujours exécuté") ;  
    System.out.println("et c'est là que l'on fermera par exemple les fichiers") ;  
}
```

et vous relancerez le programme en saisissant encore une valeur nulle pour b.

Q7 : Le bloc finally a t-il été exécuté ?

Mettez en commentaire le bloc catch et relancez le programme en saisissant encore une valeur nulle pour b.

Q8 : Le bloc finally a t-il été exécuté ?

Q9 : L'exception a t-elle été traitée ?

Pour terminer, vous allez maintenant relancer le programme en saisissant une lettre à la place d'un nombre

Q10 : Que se passe t-il ?

Q11 : Quelle exception a été lancée ?

Q12 : Corriger le problème en traitant l'exception lancée et en intégrant les lectures au clavier dans le bloc try.

Exercice n°2 :

Soit le programme oubliStupide.java suivant :

```
public class oubliStupide {  
    public static void main(String[] args) {  
        int[] tab = null;  
        System.out.println(tab[2]);  
    }  
}
```

où l'on utilise un tableau sans l'avoir créé. Mettez en place le traitement de l'exception lancée lorsque le programme s'exécute. Attention, je ne vous demande pas de corriger le programme en rajoutant par exemple `int[] tab = new int[10];`