|  |  |  |
| --- | --- | --- |
|  | **INSTITUT SUPERIEUR DES ETUDES TECHNOLOGIQUES DE NABEUL** | |
|  | Travaux Pratiques N° 9 : Les Exceptions | |
| **Classe : DSI 2** | | **Enseignante : Sana REFAI** |

## 1. Objectif

Le but de ce TP est de découvrir l’utilité des exceptions et comprendre à les intercepter et à les définir.

## 2. Les Exercices

### 2.1 Exercice 1

) Recopier dans un fichier Clavier.java la classe "Clavier" qui contient les méthodes de saisie au clavier suivantes :

- lireChaine : qui permet de saisir une chaîne.

- lireEntier : qui permet de saisir un entier.

- lireDouble : qui permet de saisir un double.

- lireFloat : qui permet de saisir un float.



**import java.io.\*;**

**public class Clavier {**

**public static String lireChaine() { String ligne\_lue=null;**

**try {**

**InputStreamReader lecteur=new InputStreamReader(System.in); BufferedReader entree=new BufferedReader(lecteur); ligne\_lue=entree.readLine();**

**}**

**catch (IOException err)**

**{ System.out.println("\*\*\*Erreur de données \*\*\*"); } return ligne\_lue;**

**}**

**public static float lireFloat() { float x=0; // valeur à lire try {**

**String ligne\_lue=lireChaine();**

**x=Float.parseFloat(ligne\_lue);**

**}**

**catch (NumberFormatException err)**

**{ System.out.println("\*\*\*Erreur de données, Réel attendu\*\*\*"); } return x;**

**}**

**public static double lireDouble() { double x=0;**

**try {**

**String ligne\_lue=lireChaine(); x=Double.parseDouble(ligne\_lue);**

**}**

**catch (NumberFormatException err)**

**{ System.out.println("\*\*\*Erreur de données, Double attendu\*\*\*"); } return x;**

**}**

**public static int lireEntier() { int x=0;**

**try {**

**String ligne\_lue=lireChaine(); x=Integer.parseInt(ligne\_lue);**

**}**

**catch(NumberFormatException err)**

**{ System.out.println("\*\*\*Erreur de données, Entier attendu\*\*\*"); } return x;**

**}**

**}**

2) Identifier les noms des exceptions qui existent.

3) Ecrire dans un autre fichier une classe TestClavier contenant une méthode main qui permet de tester ces méthodes de saisie.

### 2.2 Exercice 2

Soit la classe suivante dont la main attend un argument :

|  |
| --- |
| class Factorielle  { public static void main (String[] args)  {  int i, nbEntiers=0, factorielle=1; int ancien;  nbEntiers= Integer.parseInt(args[0]); for (i=2;i<= nbEntiers;i++)  {  ancien=factorielle; factorielle \*= i;  }  System.out.println(" Voila la factorielle des "+ nbEntiers +  " premiers entiers : "+ factorielle );  }} |

A/ Compilez et exécutez le programme Factorielle.java ci-dessus en essayant successivement :

1. de ne pas passer en ligne de commande le paramètre attendu. Que se passe t’il ?

2. de mettre un paramètre non-entier. Que se passe t’il ?

3. de mettre un paramètre entier négatif. Que se passe t’il ?

B/ modifier le programme pour que, dans chacun des cas, un message d’erreur en français soit indiqué à l'utilisateur.

Dans le cas 1, on souhaite que le programme affiche par exemple :

Donnez en paramètre la base de la factorielle

Dans le cas 2, on souhaite que le programme affiche par exemple :

La base de la factorielle doit être ENTIERE

et dans le cas 3, si le paramètre indiqué est -4 par exemple:

-4 est négatif : la factorielle d'un nombre négatif n'est pas définie

C/ Proposer une solution pour que le message du cas 3 soit affiché dans le bloc catch du cas 2

### 2.3 Exercice 3

1) Créez une classe Date qui représente une date sous la forme de 3 entiers : jour, mois et année.

Définissez un constructeur de la forme Date() qui sert à créer un objet Date en saisissant les valeurs des attributs au clavier de la manière suivante :

* le mois saisi est un entier entre 1 et 12.
* le jour saisi est un entier entre 1 et 31.

Ecrivez la classe Date en utilisant le mécanisme d'exception qui affiche un message exprimant l'erreur lorsqu’une entrée est invalide.

NB : Commencez par écrire les classes d'exceptions : JourException et MoisException.

2) Ajoutez une méthode main qui teste la classe Date avec des valeurs pouvant lever une exception.