IUT Lannion **S2 (P5)**
Département informatique

Sujet 1
(R2.03 Qual. de Dev.)

Les exceptions

Exercice 1/

Soit le code 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") ;
	}
}
```

Question 1/ Téléchargez-le depuis Moodle (Exercice 1 Code). Exécutez ce programme en saisissant une valeur 0 pour b. On a donc une erreur classique, celle d'une division par 0. Quelle exception est levée ?

Question 2/ Traitez l'exception précédente en affichant un message d'erreur de votre choix (« Alerte, c'est une division par zéro !! »).

Question 3/ Modifiez votre code de façon à ce que le message « fin du programme » soit toujours affiché, même en cas d'une division par zéro.

Question 4/ Soit le code suivant :

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

Une exception est lancée car le tableau est utilisé sans l'avoir créé (avec *new*). Mettez en place le traitement de l'exception lancée (affichage d'un message) lorsque le programme s'exécute.

Exercice 2/

Cet exercice a pour but de définir une méthode de classe qui fait la moyenne de notes (des entiers) passées en argument de la ligne de commande. La conversion en type primitif `int` de chaque argument se fera en utilisant la méthode de classe `parseInt` de la classe `Integer`. On utilisera l'exception `NumberFormatException` pour vérifier que l'argument est bien du type attendu.

Vous définirez et utiliserez une deuxième exception (`MonException`) pour vérifier que le nombre de notes passées en argument est bien différent de zéro.

Question 1/ Définissez la classe `MonException`. Le constructeur affichera un message significatif à l'écran. Prévoyez également de redéfinir `toString()`.

Question 2/ Proposez une classe munie d'une méthode de classe `moyenne` qui délivre la moyenne des notes passées par la ligne de commande. Le paramètre de cette méthode est un tableau de `String` qui contient les paramètres de la ligne de commandes. :

```
static int moyenne(String [] valeurs)…
```

Si un argument ne représente pas un entier, `l'`exception est levée mais le calcul continue avec les arguments qui suivent.

Exercice 3/

On veut écrire la méthode *saisieCorrecte* qui permet de saisir correctement un entier (en utilisant une instance de *Scanner*). Si l'utilisateur saisit une donnée dont le format n'est pas celui d'un entier, le programme l'exception *InputMismatchException* est propagée.

Question 1/ La méthode devra traiter cette erreur en fournissant une solution alternative. Un message d'erreur sera affiché avec la proposition d'effectuer une nouvelle saisie.

Question 2/ L'entier saisi doit être impérativement supérieur à 10. Compléter le code précédent pour traiter ce cas d'erreur en utilisant l'exception prédéfinie *IllegalArgumentException*. Faire une deuxième version avec une exception que vous définirez (au lieu d'utiliser *IllegalArgumentException*).

Exercice 4/

On souhaite gérer un *tirage de loto.* Le jeu du loto consiste à choisir 6 numéros compris entre 1 et 49. Un tirage aléatoire est effectué plusieurs fois dans l'année. Un joueur gagne une somme si au moins 3 numéros choisis correspondent à ceux tirés. Un tirage de loto sera muni des variables d'instance suivantes :

- numéro (*int*). C'est le numéro du tirage.
- dateT (*Date*). C'est la date du tirage (utilisez *Calendar*).
- lesNum (*Collection*). Elle contient les 6 numéros.

Il faudra prévoir les méthodes suivantes :

- un constructeur. La *Collection* est initialisée à vide.
- des méthodes pour récupérer le numéro et la date du tirage.
- une méthode pour ajouter un numéro au tirage. On fera une entrée sortie au clavier et une exception sera utilisée pour traiter le cas d'un mauvais type (ce n'est pas un entier qui est saisi par l'utilisateur). On gérera également par une exception le cas de la saisie d'une valeur entière en dehors de l'intervalle [1..49], la situation où plus de 6 numéros sont saisis et d'une valeur déjà dans le tirage.
- une méthode qui admet 6 numéros différents et qui affiche le nombre de numéros gagnants.
