

TD n°4 – Exceptions

1. Résolution

Soit le programme suivant :

```
public class Resol {
    public static void main(String[] args) {
        int k;

        try {
            k = 1 / Integer.parseInt(args[0]);
        } catch (RuntimeException e) {
            System.err.println("Runtime "+e);
        } catch (IndexOutOfBoundsException e) {
            System.err.println("Index "+e);
        } catch (ArithmaticException e) {
            System.err.println("Arithmatic "+e);
        }
    }
}
```

Lors de la compilation du programme, nous obtenons les messages d'erreur suivants :

```
Exercice2.java:11: exception java.lang.IndexOutOfBoundsException has already been caught
    catch (IndexOutOfBoundsException ex) {
        ^
Exercice2.java:14: exception java.lang.ArithmaticException has already been caught
    catch (ArithmaticException ex) {
        ^
2 errors
```

Expliquez le problème et proposez une solution.

2. Segment

Modifier directement les classes du Java Project TD3 afin de prendre en compte la **gestion des exceptions** pour les Segments :

- 1) Dans la classe Segment, **lever une exception** de type Exception dans le cas où les Points donnés pour construire le Segment sont confondus.
- 2) Quelles autres méthodes de la classe Segment peuvent alors déclencher une exception ? Ajouter les déclarations adéquates afin de **propager l'exception** jusqu'au *main* de la classe TestSegment.

Mettre à jour TestSegment afin de **traiter les exceptions**.

3. Gestion des Exceptions

La méthode `parseInt` de la classe `Integer` est spécifiée ainsi :

(<https://docs.oracle.com/javase/8/docs/api/java/lang/Integer.html>)

```
public static int parseInt(String s)
    throws NumberFormatException
```

Parses the string argument as a signed decimal integer.
The characters in the string must all be decimal digits,
except that the first character may be an ASCII minus
sign '-' ('\u002D') to indicate a negative value. The
resulting integer value is returned, exactly as if the
argument and the radix 10 were given as arguments to the
`parseInt(java.lang.String, int)` method.

Parameters:

`s` – a String containing the int representation to be parsed

Returns:

the integer value represented by the argument in decimal.

Throws:

`NumberFormatException` – if the string does not contain a parsable integer.

Créer un nouveau package et une nouvelle classe contenant uniquement la fonction `main`.

Dans le `main`, utiliser cette fonction `parseInt` pour faire la somme de tous les entiers donnés en argument de la ligne de commande, les autres arguments étant ignorés.

Exemple : java MaClasse 2 toto 3.5 12 a 1 test donnera comme résultat 15, ignorant les arguments qui ne sont pas des entiers.

4. Détection

Le programme fourni ci-dessous ne gère pas les erreurs qui peuvent se produire lors de son lancement.

- 1) Lister les erreurs d'exécution possibles dues à un lancement erroné.
- 2) Proposer des modifications du programme permettant de gérer toutes ces erreurs (en utilisant uniquement la gestion des exceptions) et rendre votre code robuste.
- 3) Vérifier que vos solutions couvrent ce qui a été demandé dans le fichier de tests `DetectionTest` donné en annexe dans Moodle.

```

public class Detect {
    private final int MAX = 99;
    private int[] tab = new int[MAX];

    > public Detect(int n) {
        for (int i=0; i<n; i++)
            tab[i] = i;
    }

    > public int sommeDesNPremiers(int n) {
        int res = 0;
        for (int i=1; i<=n; i++)
            res += tab[i-1];
        return res;
    }

    > public static void main(String[] args) {
        Detect d = new Detect(10);
        int n = Integer.parseInt(args[0]);
        int r;
        r = d.sommeDesNPremiers(n);
        System.out.print("La somme des "+args[0]+" premiers nombres ");
        System.out.println("est : "+r);
    }
}

```

5. Boite

Reprenez et terminez l'exercice de la boite du TD3.

Dans la classe `Objet` disponible dans Moodle, nous souhaitons gérer une exception dans le cas où la couleur spécifiée dans la méthode `changeCouleur` est identique à celle de l'objet courant, au lieu de simplement afficher un message d'erreur.

- Modifier la méthode `changeCouleur(Color c)` de `Objet` et lever une exception dans le cas où la couleur en paramètre est identique à celle de l'objet courant. Donner le code source complet (signature + contenu) de la nouvelle version de cette méthode.