

TP N°2

GESTION DES EXCEPTIONS

Objectif du TP

L'objectif de ce TP est de mettre en œuvre une gestion simple d'exceptions.

Mise en place du projet

Dans le projet **approche-objet**.

Exercice Reflection

La classe correspondante propose une méthode qui affiche la structure d'un objet passé en paramètre.

```
public class ReflectionUtils {  
    public static void afficherAttributs(Object obj) throws IllegalArgumentException,  
    IllegalAccessException {  
        // On commence par récupérer la classe de l'objet passée en paramètre.  
        // la classe fournit toutes les informations sur la structure d'un objet.  
        Class<?> classe = obj.getClass();  
  
        // Sur cette classe on récupère le tableau des variables d'instance  
        Field[] fields = classe.getDeclaredFields();  
  
        // On fait une boucle sur ce tableau  
        for (Field field : fields) {  
            // On force l'accès à l'attribut depuis une classe externe.  
            // Un peu comme si je forçais l'attribut en lisibilité public.  
            field.setAccessible(true);  
  
            // Affichage  
            System.out.println("La valeur de l'attribut " + field.getName() + " est " +  
field.get(obj).toString());  
        }  
    }  
}
```

- 1) Créez un nouveau package **fr.diginamic.testexceptions**
- 2) Copiez la classe **ReflectionUtils** dans ce package
- 3) Créez une classe exécutable **TestReflectionUtils**
 - a. Dans cette classe, créez une instance de **Ville** et invoquez la méthode **afficherAttributs** de **ReflectionUtils** avec cette instance de **Ville**.

- 4) Que se passe t'il si vous invoquez cette méthode avec **null** ? Comme vous le constatez cette méthode n'est pas robuste. Nous allons y remédier en gérant une exception dans le cas où on passe **null** en paramètre de cette méthode :
- Créez une exception **ReflectionException** qui hérite d'**Exception**
 - Modifiez la méthode **afficherAttributs** afin qu'elle jette l'exception **ReflectionException** dans le cas où le paramètre de la méthode est null
 - Testez maintenant votre classe **TestReflectionUtils** afin de vérifier que cela fonctionne correctement.

TP RecensementPopulation

Dans ce TP, nous allons mettre en place une gestion des exceptions sur **l'application de recensement**.

L'objectif est de rendre l'application plus robuste et aussi de fournir des messages intelligibles lorsque l'utilisateur saisit des informations incohérentes.

Mise en place

- 1) Forkez le dépôt suivant :

Pour rappel le « fork » permet de copier ce dépôt sur votre espace GitHub.

<https://github.com/DiginamicFormation/java-poo-j6.git>

- 2) Cloner ensuite ce dépôt dans votre workspace.

Comme vous pouvez le constater dans l'application, on commence par charger **la liste des villes** puis on crée également **un scanner** pour les interactions futures avec l'utilisateur (question/réponse).

Ensuite, **chaque option de menu** est traitée dans une **classe de service** à laquelle on passe tout ce dont elle a besoin pour traiter le cas fonctionnel : la liste de villes et le scanner.

Dans ce TP, on va commencer par s'intéresser à la classe **RecherchePopulationBorneService** qui est appelée lorsque l'utilisateur clique sur 4:

- dans ce cas d'utilisation, on pose 3 questions à l'utilisateur
 - quel département l'intéresse
 - quel est le minimum de population à rechercher
 - quel est la max de population à rechercher
- Une fois que les 3 questions sont renseignées, on affiche toutes les villes du département dont la population est comprise entre le min et le max.

Travaux à Réaliser :

- **Envoyez une exception avec un message clair dans tous les cas suivants :**
 - si l'utilisateur saisit une lettre au lieu de chiffres pour le min ?
 - si l'utilisateur saisit une lettre au lieu de chiffres pour le max ?
 - si l'utilisateur saisit un min < 0 ou un max < 0 ou un min $>$ max ?
 - si l'utilisateur saisit un code département inconnu
- Dans la classe **Application**, **catchez les exceptions** pour afficher le message d'erreur.
- Ensuite, **mettez en place des contrôles dans les autres classes de services** dans le cas où le nom de la ville, le code du département ou le code de la région n'existe pas.

Commitez vos développements sur GitHub