

TD : GESTION DES EXCEPTIONS

EXERCICE 1 : Contrôle et capture des exceptions

Soit les deux classes d'exception suivantes :

```
class Exc1 extends RuntimeException { .... }
class Exc2 extends IOException { .... }
```

Soit la classe suivante :

```
public class TestException {
    public void f1() { throw new Exc1(); }
    public void f2() { throw new Exc2(); }
    .....
}
```

- 1) Expliquer pourquoi f1() se compile sans problème, tandis qu'une erreur de compilation se produit lors de l'analyse de f2() ?
- 2) Proposer une solution pour résoudre ce problème de compilation.

Exercice 2 : Exceptions personnalisées avec passage de paramètre

Réaliser une classe **EntNat** permettant de gérer des entiers naturels (positifs ou nuls) et disposant :

- d'un constructeur avec un argument de type int ; il générera une exception de type *ErrConst* si la valeur de son argument est négative ;
- un accesseur en lecture getN() qui fournira sous forme d'un int la valeur encapsulée dans un objet de type EntNat ;
- un accesseur en écriture setN() qui modifiera la valeur de l'entier naturel grâce à un int passé en paramètre ; cette méthode générera une exception de type *ErrModif* si la valeur passée en paramètre est négative ;
- une méthode decremente() qui décrémente de 1 un objet EntNat ; cette méthode devra pouvoir lever une exception de type *ErrModif* ;
- une méthode de classe – statique donc – decremente(EntNat e) qui décrémente de 1 l'objet passé en paramètre (*c'est juste pour que vous travaillez sur les méthodes de classe, il serait en effet normal d'en faire une méthode d'instance ...*)

Écrire une méthode **main** qui utilise les méthodes de la classe EntNat, en capturant les exceptions susceptibles d'être générées.

Exercice 3 : Exceptions personnalisées

Implémentez la classe « Projet » caractérisée par :

- le nom du projet (String),
- le sujet du projet (String)
- la durée (int)

Le programme doit garantir que :

- Seuls les projets dont le nom et le sujet n'excède pas les 20 caractères seront validés;
- la durée du projet soit bien un entier positif.
- Vous introduirez pour cela deux classes d'exceptions personnalisées `WrongDurationException` et `NameTooLongException`

On se contentera du constructeur par défaut pour la classe `Projet`.

On implémentera la méthode `readInt()` retournant un entier. Cette méthode est susceptible de lever une exception si le type de la donnée saisi au clavier est incorrect ou si la durée a une valeur négative.

Pour lire le nom et le sujet du projet, on doit implémenter une méthode `readString()` qui retourne la chaîne lu au clavier. Cette méthode peut lever une exception si la chaîne dépasse les 20 caractères.

`readInt()` et `readString()` laisseront se propager les erreurs qui seront capturées au niveau de la méthode `readProjet()`.

La méthode `readProject` fait appel à `readInt()` pour lire la durée du projet et `readString()` pour lire le nom et le sujet du projet. Cette méthode redemande les données jusqu'à ce qu'elles soient correctes.

Réaliser une classe `Test` pour tester votre code.