

TP : Rappels sur Java

Partie 1 : Les exceptions

Exercice 1 :

Le but de cet exercice est de simuler la gestion d'un système de gestion de bibliothèque en utilisant des exceptions.

- 1- Créez une classe Livre avec les attributs titre, auteur, année de publication.
- 2- Créez une classe Bibliotheque qui peut stocker un certain nombre de livres.
- 3- Implémentez des méthodes pour ajouter et supprimer des livres de la bibliothèque.
- 4- Ajoutez de la logique pour vérifier si la bibliothèque est pleine (par exemple, limite de 100 livres).
- 5- Créez une exception personnalisée BibliothequePleineException qui sera lancée si quelqu'un essaie d'ajouter un livre lorsque la bibliothèque est pleine.
- 6- Écrivez un programme Java qui utilise la classe Bibliotheque pour ajouter des livres et capture les exceptions en cas de bibliothèque pleine.

Exercice 2 :

Le but de cet exercice est de créer une liste de tâches (to-do list) avec la possibilité d'ajouter et de supprimer des tâches tout en gérant les exceptions.

- 1- Créez une classe TodoList qui contient une ArrayList de chaînes (représentant les tâches).
- 2- Implémentez une méthode pour ajouter une tâche à la liste. Dans cette méthode, vérifiez si la tâche est vide (chaîne vide) et lancez une exception personnalisée TacheVideException si c'est le cas.
- 3- Implémentez une méthode pour supprimer une tâche de la liste. Dans cette méthode, vérifiez si l'indice spécifié pour la suppression est valide (dans les limites de la liste) et lancez une exception personnalisée IndiceInvalideException si ce n'est pas le cas.
- 4- Écrivez un programme Java qui utilise la classe TodoList pour ajouter et supprimer des tâches, en capturant les exceptions au besoin.

Partie 2 : Gestion de fichiers et flux d'entrée/sortie

Exercice 3 :

Écrivez un programme Java qui permet de créer un fichier newfile.txt sur votre bureau. Le programme affichera 'Fichier créé avec succès' si l'opération de création se déroule sans problème. Sinon, il affichera 'Le fichier existe déjà' s'il découvre que le fichier existe déjà, et il insérera une phrase dans le fichier.

Vous pouvez utiliser la classe `FileWriter` pour ouvrir un flux d'écriture vers le fichier en utilisant son constructeur, ainsi que sa méthode `write()` pour effectuer l'écriture.

```
FileWriter( String nom_du_fichier, boolean append )
```

- Si la valeur du boolean `append` est mis à `false`, le contenu du fichier sera écrasé par le nouveau contenu.
- Sinon, le nouveau contenu sera ajouté à l'ancien.

Exercice 4 :

Écrivez un programme Java qui copie le contenu d'un fichier dans un autre à l'aide de `FileInputStream` et `FileOutputStream`.

Le programme doit prendre deux chemins de fichiers en entrée et créer une copie du fichier source dans le fichier de destination.