

TD3 Exceptions

Exercice 1 :

Ecrire un programme qui va lire un fichier texte. Gérer une éventuelle exception qui peut apparaître. (Eclipse vous aidera)
Attention à bien fermer l'objet de lecture (InputStream ou Scanner)
Que se passe t'il si le fichier n'existe pas ???

Exercice 2 :

Ecrire un programme qui fait une série de 10 division de 2 entiers générés aléatoirement entre 0 et 20 avec la classe Random.
Attention à bien gérer la division par 0 !

Problème :

Une banque a besoin d'un logiciel pour gérer les transaction par virement de ses clients.

Une transaction consiste à débiter une somme sur un compte client, et d'en créditer un autre. Si une erreur devait survenir, nous devons remettre tous les comptes dans leur état initial.

La banque (le client) veut un POC (Proof of concept) pour la fin de journée sur 3 comptes.

Un compte comprend :

- un numéro
- un nom de détenteur
- un solde.

Ecrire un programme permettant d'initialiser 3 comptes :

- Le premier avec le numéro 10029344 au nom de Maurice Dupond qui contient 10108€
- Le second avec le numéro 30499450 au nom de Fabrice Millot qui contient 1304€
- Le troisieme avec le numéro 4059963 au nom de Nora Excillia qui contient 300400€

Réaliser une série de 10 opérations entre les comptes.

Contrainte :

Avant et après chaque opération de débit et de crédit, il faut appeler ce code là pour simuler une éventuelle erreur (déconnexion de réseau, système d'une autre banque qui ne répond pas, ...) :

```
private static void simulerErreur() throws VirementException {
    Random r = new Random();
    if(r.nextBoolean()){
        throw new VirementException("Une erreur est survenue lors de
la transaction");
    }
}

private static class VirementException extends Exception {
    public VirementException(String string) {
        super(string);
    }
}
```

Etape 1 : Ecrire la classe Compte contenant les données énoncées ci-dessus, le ou les constructeurs et les accesseurs

Etape 2 : Dans une classe « utilitaire » VirementUtils (avec des méthodes statiques), copier le code de la méthode simulerErreur() et de la sous-classe VirementException (qui étend Exception)

Etape 3 : écrire une méthode executerVirement (publique et statique) qui prend 2 comptes et réalise le virement de la manière suivante :

simuler une erreur
débit source
créditer cible
simuler une erreur

Etape 4 : Réaliser des virements dans une classe avec une méthode main.