

# Activité intégrative (B2)

- [Exceptions](#)

# Exceptions

Comme vu dans le cours sur la programmation orientée Java, on peut créer des exceptions, les catcher (avec try-catch) et utiliser un bloc “finally” (avec try-catch-finally).

Cependant ici on va surtout voir comment utiliser try-with-resources qui permet d’ouvrir un flux de donnée et d’automatiquement le fermer par après. *Voir le cours sur les structures de données pour mieux comprendre pourquoi*

Ainsi pour utiliser le try-with-resources, il faut utiliser une classe implémentant l’interface `Closeable`, on peut également créer notre propre classe pour cela pour créer notre propre type de ressource.

L’interface `Closeable` est une interface n’ayant comme seule méthode `void close()` pour fermer le flux.

Ensuite pour utiliser le try-with-resources on peut le faire comme ceci :

```
// Ici un FileReader (implémentant Closeable) va être ouvert
try (FileReader reader = new FileReader("my_file.txt")) {
    // On essaye de lire le fichier pour mettre les caractères dans un array
    reader.read(this.characters);
} catch (IOException e) {
    // Si une IOException arrive, on log cela dans la console
    logger.error(e.getMessage());
}
// Une fois cela fait, le flux est automatiquement fermé
```