Exceptions Java: Dompte les Bugs et Deviens un Ninja du Code!

Le Menu du Jour:

Les Exceptions: C'est quoi ce bazar?

Try, Catch, Finally: Le trio infernal (mais indispensable!)

Throw & Throws: Lancer et Propager la Panique (avec contrôle!)

Exceptions Perso: Crée tes propres erreurs sur mesure!

Inventaire: Un exemple concret pour tout comprendre

Exceptions: Quand le Code Part en Vrille!

- Exception: Un événement inattendu qui casse le déroulement normal du programme
  - Comme un caillou dans une chaussure ou une panne d'essence sur l'autoroute
- Il faut savoir les anticiper et les gérer pour éviter le crash total!

Exceptions: Les Types de Catastrophes

- RuntimeException: Les erreurs 'unchecked', pas besoin de les déclarer
  - Comme oublier de vérifier si une variable est nulle
- Exception: Les erreurs 'checked', il faut les gérer ou les déclarer
  - Comme essayer de lire un fichier qui n'existe pas

Try/Catch/Finally: Le Kit de Survie!

- try { ... } : Le bloc de code où l'erreur peut arriver
- catch (Exception e) { ... } : Le bloc qui gère l'erreur si elle se produit
- finally { ... } : Le bloc qui s'exécute TOUJOURS, qu'il y ait eu une erreur ou pas (pour fermer les fichiers, etc.)

Try/Catch/Finally: Exemple de Code

- try {
- int result = 10 / valeur; //Attention, division par zéro possible!
- } catch (ArithmeticException e) {
- System.out.println("Division par zéro !"); //On gère l'erreur
- } finally {
- System.out.println("Fin du bloc try/catch"); //On fait le ménage!
- }

Throw & Throws: Balance la Sauce!

- throw new Exception() : Balance une exception à la face du code!
  - Comme dire: 'Je suis pas capable de gérer ça, débrouillez-vous!'
- throws Exception : Indique que la méthode peut lancer une exception
  - Comme mettre un panneau 'Attention, risque de chute de pierres!'

Throw & Throws: Exemple de Code

- if (or < 0) {
- throw new IllegalArgumentException("Or négatif non autorisé"); //On balance l'erreur!
- }
- public void chargerInventaire(String fichier) throws IOException { //Attention, ça peut planter!
- // peut lancer IOException
- }

Exceptions Perso: Fais Ton Propre Danger!

- Pour créer des exceptions qui correspondent à ton code (et pas juste des erreurs génériques)
  - Comme créer une exception 'Pas assez de mana' dans un jeu vidéo

Exceptions Perso: Exemple de Code

- public class InventaireException extends Exception { //On crée une nouvelle famille d'exceptions
- public InventaireException(String message) { super(message); }
- }
- public void equiperObjet(Joueur joueur, Item item) throws InventaireException { //On précise qu'on peut lancer cette exception
- if (joueur.getNiveau() < item.getNiveauRequis()) {
- throw new InventaireException("Niveau insuffisant pour équiper l'objet"); //On balance l'exception perso!
- }

À toi de jouer! - Division Risquée

- Crée une méthode `public static int division(int a, int b)`
- Elle doit retourner a/b
- Mais attention! Gère l'ArithmeticException si b vaut 0
- Affiche un message d'erreur sympa dans le catch
- Retourne 0 si une exception se produit

À toi de jouer! - Fichier Fantôme

- Crée une méthode `public static String lireFichier(String chemin) throws IOException`
- Elle doit utiliser `new FileReader(chemin)` pour lire un fichier
- Elle doit déclarer `throws IOException` pour propager les erreurs

À toi de jouer! - Niveau Trop Faible

- Crée une exception perso: `class NiveauInsuffisantException extends Exception`
- Dans `public void equiperObjet(Joueur j, Item i)`, lance cette exception si le joueur n'a pas le niveau

À toi de jouer! - Fermeture Impossible

- Imagine un code qui ouvre un fichier (on ne va pas le faire en vrai)
- Utilise `try` pour ouvrir le fichier et lire une ligne (fais juste un System.out.println("Lecture..."))
- Utilise `finally` pour fermer le fichier (même si ça plante!) (fais juste un System.out.println("Fermeture..."))

À toi de jouer! - Documentation au Top!

- Retourne dans la méthode `retirerObjet` du cours précédent
- Ajoute une JavaDoc pour expliquer qu'elle peut lancer une exception `InventaireException`
- Explique dans quel cas cette exception est lancée (si l'objet n'est pas dans l'inventaire)

Besoin d'aide? Ressources Utiles

- Gestion des exceptions (Oracle) : https://docs.oracle.com/javase/tutorial/essential/exceptions/
- Créer exception personnalisée (Baeldung) : https://www.baeldung.com/java-custom-exceptions