

« TuniProd »

Objectifs :

- Dissocier la détection d'une anomalie de son traitement.
- Séparer la gestion des anomalies du reste du code.

Notre application rencontre certaines circonstances exceptionnelles qui peuvent compromettre la poursuite normale de son exécution par exemple des données incorrectes. Pour cela on devra utiliser un mécanisme pour assurer la bonne exécution et omettre certaines situations.

Pour cela, on va reprendre les deux classes `Produit` et `Magasin`

1. Dans la méthode `ajouterProduit` qui ajoute un produit dans le magasin, enlevez les tests qui vérifient que le magasin n'est pas plein.
2. Modifier la taille maximum du tableau produit à 2
3. Tester l'ajout dans la classe de test des 3 produits
4. Qu'est vous remarquez ?
5. corrigez l'erreur en proposant une solution
6. Ecrivez votre propre classe d'exception **`MagasinPleinException`**, pour ce type d'exception. En effet, la méthode **`ajouterProduit()`** devra utiliser cette nouvelle exception.
7. On voulait interdire, l'ajout d'un produit ayant un prix négatif, en effet, nous devons renvoyer l'exception levée à une classe **`PrixNegatifException`** qui va la gérer.