

# Test et qualité – TD 1

## Exercice 1

Créez une interface Calcul avec une fonction *double calculer(double ... args)*.

1. Créez une classe implémentant Calcul qui doit faire le calcul suivant :  $\sqrt{1/(x^2-1)}$
2. Créez une classe de test permettant de valider fonctionnellement votre classe. Pensez à étudier les différentes classes d'équivalence.
3. Créez une classe implémentant Calcul qui doit faire le calcul suivant :  $\sqrt{(y*(z-1)/(x^2-1))}$
4. Créez une classe de test permettant de valider fonctionnellement votre classe. Combien de tests fonctionnels devrait il y avoir ? Utilisez Pairwise pour limiter le nombre de tests et implémentez les.

## Exercice 2 – Tests fonctionnels ↔ Tests d'intégration

Créez une interface TriTableau avec une fonction `Integer[] trier(Integer[] t)` dont la fonctionnalité principale est de trier les éléments d'un tableau dans l'ordre croissant

1. Définir une classe abstraite permettant de valider les futures implémentations de l'interface TriTableau d'un point de vue fonctionnel.
2. Créez une classe TriBulleTableau implémentant un tri bulles sur un tableau d'entiers.
3. Vérifiez que la classe que vous avez implémenté satisfait vos tests fonctionnels.
4. Codez une autre implémentation d'un tri (de votre choix) et vérifiez que votre nouvelle implémentation satisfait également les cas de tests prévus

[En vous appuyant sur les réalisations ci-dessus, implémentez et testez un tri de listes d'entiers.](#)

## Exercice 3 – Validation d'un code existant

Récupérez sur Célène le fichier exercice3.zip. Essayez de valider la fonction implémentée dans la classe OutilsListe. Quel est votre verdict ?