La méthodologie autour des tests
TD 1: Les tests unitaires

Exercice 1 : Avenger

1/ Créer une classe de test pour la classe Thanos

2/ Écrire un test pour le constructeur vérifiant que Thanos a bien le nombre de pierre qu'on lui donne et qu'il n'a pas réussi sa mission

- Tester en initialisant thanos avec 0 pierre.
- Tester en initialisant thanos avec 5 pierre.

3/ Tester la méthode gagnePierre.

- Tester en initialisant thanos avec 0 pierre.
- Tester en initialisant thanos avec 5 pierre.
- Tester avec RepeatedTest(6)

4/ Tester le retour de la méthode claquementDeDoigts()

- Tester en initialisant thanos avec 0 pierre avec un nbPopulation à 7 700 000.
- Tester en initialisant thanos avec 5 pierre avec un nbPopulation à 7 700 000.

5/Tester la méthode toString

- Tester en initialisant thanos avec 0 pierre.

6/ Réinitialisez deux thanos ayant 0 pierre et testons l'égalité des deux.

a- Sans modifier le code de la class

b- Faite généré par l'IDE la méthode toEquals()

7/Écrire ces tests avec la librairie AssertJ

Exercice 2: Harry Potter

1/ Tester que à l'initialisation, il y a bien 4 maison mais qu'il n'y a pas d'élèves.

2/ Testons inscriptionEleve ("Lucius Malefoy", Serpentard);Utiliser DisplayName pour afficher "Test de l'inscription de Lucius Malefoy"

3/Tester arriveDesHeros

4/Tester la méthode findMaison suite à l'arrivé des héros avec `@ParameterizedTest @ValueSource. Faire afficher le message Test de la maison de 'Harry Potter'`

5/ Refactoriser en utilisant BeforeEach

6/ Écrire ces tests avec la librairie AssertJ

Exercice 3:Les mondes de Ralph

Un bug s'est glissé dans le code du jeu Vaneloppe. Je l'ai remarqué grâce au test unitaire du développeur. Maintenant il faut le corriger.
a- Trouver la ligne qui ne fonctionne pas avec l'aide du débogueur.
b- Comment aurait il pu faire pour que le débogage est été plus rapide?