TP Tests : Le jeu de la boule

# Règle du jeu :

https://fr.wikipedia.org/wiki/Jeu\_de\_la\_boule\_%28jeu\_de\_casino%29

Le jeu de la boule est un jeu de casino similaire à celui de la roulette, mais avec les chiffres de 1 à 9. Il est possible de miser rouge, noir, manque, passe, pair ou impair.

Les chiffres 1, 3, 7, 9 sont impairs. Les chiffres 2, 4, 6, 8 sont pairs. Les chiffres 1, 3, 6, 8 sont noirs. Les chiffres 2, 4, 7, 9 sont rouges. Les chiffres 1, 2, 3, 4 sont manques. Les chiffres 6, 7, 8, 9 sont passes.

Le chiffre 5 n'est ni pair, ni impair, ni manque, ni passe, ni rouge, ni noir. Si le 5 sort, la mise jouée sur une chance simple perd.

Si le numéro misé sort, le joueur gagne 7 fois la mise sinon la mise est perdue.

Contrairement à la roulette, il n'est pas possible de miser plusieurs numéros avec un seul jeton.

# Tache 1 : Implémentation d’une classe en TDD

La classe BouleGameImpl implémente déjà la classe BouleGame avec une méthode bet. Cette méthode doit tirer au sort un nombre et évaluer si le pari soumis est gagnant en donnant le ratio de gain du joueur (0, 1 ou 7).

La classe de test BouleGameImplTest fournit les premiers cas de test.

Piste : pour savoir comment tester une classe qui utilise un générateur de nombre aléatoire, utiliser un mock. La documentation de mockito : <http://mockito.org/>

# Tache 2 : Compléter vos tests

* Que se passe-t-il si on passe null en paramètre de la méthode ?
* Vos tests sont-ils exhaustifs ? tous les cas sont-ils testés ? comment le savoir => target/site/jacoco/index.html
* Compléter vos tests pour avoir 100% de couverture.

# Tache 3 : Tests fonctionnelles avec la classe UserBetManagerImpl

1. Supprimer l’annotation @Ignore sur la classe de test UserBetManagerImplTest.
2. Compléter la classe UserBetManagerImpl pour que le test UserBetManagerImplTest passe.
3. Compléter les scénarii de test (feature) pour couvrir le code.

Documentation : https://cucumber.io/