TP Noté - Testing

Dans ce TP nous allons mettre à profil nos connaissances pour réaliser le test unitaire d’un système dit « chaotique ».

Le rendu se fait sur clef USB, dans un zip contenant le code source complet, sous forme « NOM.Prenom.zip ».

# But

Nous utilisons un système chaotique pour générer des nombres aléatoires. Un système chaotique est, comme son nom l’indique, très variant et souvent imprévisible. Source Wikipedia :

Système dynamique chaotique

Un système dynamique est dit **chaotique** si une portion « significative » de son espace des phases présente simultanément les deux caractéristiques suivantes :

* le phénomène de sensibilité aux conditions initiales,
* une forte récurrence.

# Suite chaotique (4pts)

Avant de réaliser les tests unitaires, il convient de créer le programme qui va réaliser notre système chaotique. Nous choisirons la suite chaotique suivante :

$$U\left(n+1\right)= 0.5\*(U\left(n\right)- \frac{1}{U\left(n\right)})$$

1. Créer un package « com.suite »
2. Créer une classe « Chaos »

Cette classe doit avoir un constructeur qui accepte un paramètre : la valeur de départ : $U(0)$

Elle doit fournir deux fonctions. La classe n’utilise que les types « int » et « double » :

1. Une fonction « getIteration » qui prend en paramètre un entier « iter » et qui renvoie la valeur $U(iter)$ calculée par la suite.
2. Une fonction « checkIteration » qui prend en paramètre un double (le résultat de $(quelque\\_chose)$ ), on cherche à retrouver la valeur de « quelque\_chose » initiale (soit l’opération inverse de getIteration). On cherchera dans la tranche 0 <= quelque\_chose <= 1000 pas plus pour ne pas surcharger le système. En cas d’erreur (quelque\_chose n’est pas trouvé), on retourne -1 (ce qui indiquera une erreur, puisque le résultat « bon » sera un entier dans la tranche 0 à 1000 inclus).

Avant d’aller plus loin, nous vous conseillons de regarder le comportement de Chaos, avec les valeurs :

1. $U\left(0\right)= 2 et U\left(0\right)= -2$
2. $U\left(0\right)=6 et U\left(0\right)= -6$

Mettez en commentaire quelque part dans le code (en indiquant la mention « question 1 »), vos appréciations sur le comportement de la suite lorsque le signe est inversé.

# Tests (10pts)

On vous demande de créer un package « test.suite » qui sera utilisé pour réaliser les tests unitaires.

Une fois ce package fait, vous créerez les tests unitaires.

Proposez (dans un commentaire « question 2 »), au moins deux méthodes différentes pour réaliser les tests unitaires assurant la meilleure qualité de code possible.

Vous êtes libre de choisir les tests à faire, mais vous devez, en haut de chaque classe de tests, expliquer le choix des tests fait, et leurs implications. De même, vous devez expliquer pourquoi certains tests ne sont pas faits, et quelle est l’incidence sur la fiabilité du système (sans incidence, faible, critique…).

# Suite chaotique (v2) (6pts)

On cherche à démontrer l’un des phénomènes des suites chaotiques : leur forte divergence en fonction de la précision du système. Créez une classe Chaos2, copie conforme de Chaos précédemment créé, mais qui cette fois, travaille en « int » et « float ».

Mettez en commentaire quelque part dans le code (en indiquant la mention « question 3 »), le comportement de Chaos2 face à Chaos.

Proposez un test unitaire permettant, à l’aide de Chaos2, de comparer le comportement avec Chaos1 sur les 20 premières itérations. On vous conseille de réaliser un commentaire pour bien expliquer votre démarche, puis, de faire le code qui réalise les tests unitaires.