

TP2 : Paquetages, Exceptions et Javadoc

application : suites récursives

Le but de ce TP est de réaliser un programme Java permettant de manipuler des suites récursives. Les suites pourront être soit des suites arithmétiques, soit des suites géométriques.

Le code écrit pour les exercices de ce TP devra être remis sur la page Moodle de ce module d'ici le début de la prochaine séance. Il est recommandé de rédiger un petit rapport (fichier texte) contenant les réponses aux questions posées, ainsi que tout ce qui vous semblera pertinent : remarques sur votre code, stratégie d'implémentation, difficultés rencontrées, ...

1 Rappels mathématiques

Les termes de la suite sont notés $(u_n)_{n \in \mathbb{N}}$. Le premier terme a pour indice 1.

Suite arithmétique

On suppose donnés les valeurs du premier terme u_1 et la raison r (valeur de la progression).

Les termes suivants sont calculés de façon récursive : $u_{i+1} = u_i + r$

Calcul du $n^{\text{ème}}$ terme : $u_n = u_1 + (n - 1) \cdot r$

Calcul de la somme des n premiers termes : $u_1 + u_2 + \dots + u_n = n \cdot \frac{u_1 + u_n}{2}$

Suite géométrique

On suppose donnés les valeurs du premier terme u_1 et de la raison q (valeur de la progression).

Les termes suivants sont calculés de façon récursive : $u_{i+1} = qu_i$

Calcul du $n^{\text{ème}}$ terme : $u_n = u_1 \cdot q^{n-1}$

Calcul de la somme des n premiers termes : $u_1 + u_2 + \dots + u_n = \begin{cases} n \cdot u_1 & \text{si } q = 1 \\ u_1 \cdot \frac{1 - q^n}{1 - q} & \text{sinon} \end{cases}$

2 Programmation Java

2.1 Documentation : javadoc

Il vous est demandé de commenter votre programme et générer les pages HTML correspondantes en utilisant *javadoc*. Prenez l'habitude de commenter vos classes et méthodes au fur et à mesure.

2.2 Code initial

Écrire un programme Java qui comporte :

- une classe abstraite *Suite* ;
- les classes concrètes *SuiteArithm* et *SuiteGeom* ;
- une classe *Test* comportant la méthode *main*.

2.2.1 Classe « Suite »

Cette classe comporte

- les attributs
 - premier* 1er terme de la suite (u_1);
 - pas* valeur de la progression entre deux termes successifs de la suite (noté r ou q dans les rappels mathématiques).
- un constructeur *Suite(int, int)* qui initialise les 2 attributs *premier* et *pas*.
- les méthodes abstraites
 - *int valeurAuRangN(int)* qui permet de calculer u_n , n étant transmis en argument;
 - *int sommeAuRangN(int)* qui permet de calculer la somme des n premiers termes, n étant transmis en argument.

2.2.2 Classes « SuiteArithm » et « SuiteGeom »

Ces deux classes dérivent de *Suite* et

- ont chacune un constructeur;
- comportent le corps pour les méthodes *valeurAuRangN* et *sommeAuRangN*.

Indications :

- la classe *Math* a une méthode *static double pow(double a, double b)* qui renvoie a^b .
- faire attention aux arrondis :
 - int i = 3/2 * 8*; affecte la valeur 8 à i alors que
 - int i = 3 * 8/2*; affecte la valeur 12 à i .

2.2.3 Classe « Test »

Cette classe comporte la méthode *main* avec

- déclaration de deux suites *s1* et *s2*;
- instanciation de *s1* comme suite arithmétique et *s2* comme suite géométrique;
- appels pour *s1* et *s2* aux méthodes *valeurAuRangN* et *sommeAuRangN*.

2.3 Constitution de paquetages

1. Mettre les classes *Suite*, *SuiteArithm* et *SuiteGeom* dans un paquetage nommé *paquetSuite*. La classe *Test* reste dans le paquetage anonyme. Compiler et exécuter le programme.
2. Mettre la classe *Test* dans un paquetage *paquetMain*. Compiler et exécuter le programme.

Vous noterez vos observations (comment compiler un paquetage, d'où l'exécuter, etc.) dans vos notes.

2.4 Gestion des exceptions

1. Modifiez le code de votre programme afin de vérifier que le rang transmis en argument dans les méthodes *valeurAuRangN()* et *sommeAuRangN()* est bien positif. Pour ce faire, vous pouvez, dans l'ordre :
 - (a) utiliser la classe *Exception*,
 - (b) créer votre propre classe *SuiteException*, sous-classe de *Exception*,
 - (c) mettre votre classe *SuiteException* dans un paquetage *paquetException*.
2. Complétez votre documentation avec *@throws* ou *@exception*. Générez la nouvelle documentation.
3. Complétez la classe *SuiteArithm* avec la méthode *int calculRang(int)*, où l'argument correspond à un certain u_n , et le résultat renvoyé par la méthode est la valeur de l'indice n . Exemple : pour une suite arithmétique s , avec $u_1 = 3$ et $r = 4$, $s.calculRang(19)$ renverra 5, car $u_5 = 19$ pour cette suite. Envisagez le cas où l'argument transmis ne correspond pas à un terme u_n .

3 AGL : diagrammes de classes

Environnement

Pour ce travail, on vous demande d'utiliser un logiciel d'AGL (Atelier de génie logiciel), par exemple ArgoUML. Une fois téléchargé, vous pourrez extraire l'archive et lancer l'application grâce à la commande `java -jar argouml.jar`.

Exercice

Construire le diagramme de classes correspondant à l'exercice précédent.
Exporter vos classes en schéma au format PNG et analysez-le.
Exporter vos classes en code Java et un autre langage de votre choix¹. Regarder le résultat.
Saisir les commentaires et ré-exporter le code source.
Comparer avec votre implémentation et faites d'éventuelles modifications afin d'obtenir un code exporté cohérent avec votre implémentation.

1. Faites cette manip dans un dossier séparé afin de ne pas écraser vos classes.