

# Calculatrice – version 3<sup>1</sup>

## Compétences visées :

- Maîtrise de la configuration de son environnement de travail et du processus de développement logiciel d'une application développée en Java.
- Maîtrise des packages, des exceptions, des interfaces et des mots-clés en Java.
- Apprentissage des notations UML pour l'implantation Java.

- 1.1. Configurez votre environnement de travail (JDK et Eclipse) pour réaliser un projet Java, nommé Calculatrice-v3.
- 1.2. Dans votre projet Calculatrice-v3, vous allez développer une calculatrice qui supporte un nombre extensible d'opérations de types différents (binaire, unaire et ensembliste). Le diagramme de classes de cette nouvelle version de la calculatrice est donné dans la Figure 2.

Le nom retourné par la méthode `getNom()` pour les différentes opérations est donné dans le Tableau 1.

Opérations binaires		Opérations unaires		Opérations ensemblistes	
Ajouter	+	Cosinus	cos	Maximum	max
Soustraire	-	Sinus	sin	Minimum	min
Multiplier	*	Tangente	tan		
Diviser	/	Valeur absolue	abs		

Tableau 1 : Nom des opérations de la calculatrice.

Pour ne pas surcharger le diagramme de classes, le Tableau 2 récapitule l'ensemble des messages d'erreurs qui seront levés par les différentes classes.

Classes	Méthodes	Messages
Calculatrice	<code>calculer()</code>	Opération inexistante dans la calculatrice
OperationBinaire	<code>calculer()</code>	Nombre d'opérandes != 2 pour une opération binaire
OperationUnaire	<code>calculer()</code>	Nombre d'opérandes != 1 pour une opération unaire"

<sup>1</sup> Ce sujet a été conçu initialement en collaboration avec Dr. Yoann Maurel, enseignant-chercheur à l'Université de Rennes 1.

OperationEnsembliste	calculer()	Ensemble vide pour une opération ensembliste
Diviser	doCalculer()	Division par zéro

Tableau 2 : Messages d'exception levés dans les différentes classes.

A titre exceptionnel, il a été représenté, dans la classe Calculatrice, les deux constructeurs.

Vous devrez, à la fin, tester votre développement grâce à une classe Calculateur (non représentée dans le diagramme), qui devra vous permettre d'évaluer plusieurs calculatrices avec différents types d'opérations.

De plus, votre projet doit comprendre un fichier build.xml, qui permet de compiler, packager et exécuter votre programme.

**Suite au développement, répondez à l'ensemble de ces questions :**

- 1.2.1.A quoi sert l'interface Opération ? A quoi servent les classes abstraites OperationBinaire, OperationUnaire et OperationEnsembliste ?
- 1.2.2.Quel est l'intérêt d'utiliser la visibilité « protected » pour les méthodes doCalculer des classes abstraites OperationBinaire, OperationUnaire et OperationEnsembliste ?
- 1.2.3.Dans la classe CalculatriceException, vous avez dû ajouter un attribut serialVersionUID. Si ce n'est pas le cas, vous avez un warning. Cliquez sur le triangle à gauche de la déclaration de la classe, puis choisissez « Add generated serial version ID » (Figure 1).

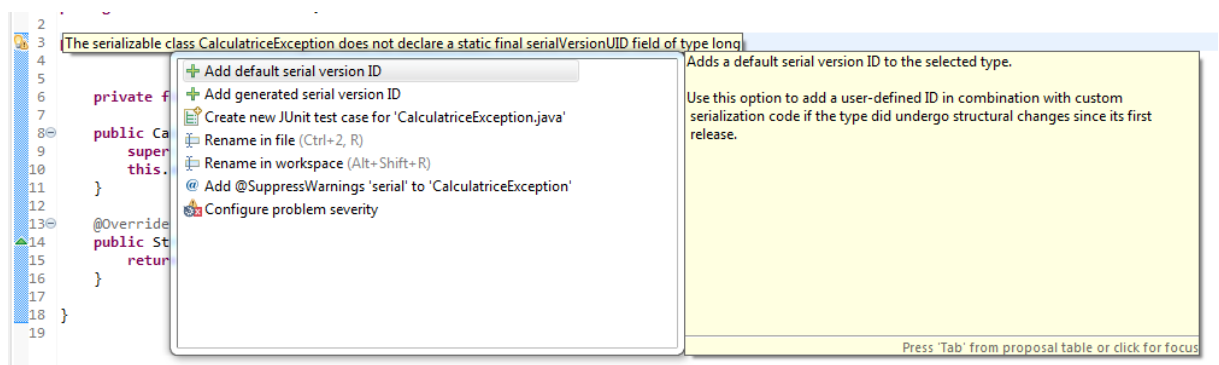


Figure 1 : Ajout de l'attribut serialVersionUID.

A quoi sert l'attribut serialVersionUID ? D'où vient-il ?

- 1.2.4.D'un point de vue Génie Logiciel, après avoir développé trois versions de la calculatrice, quel est l'intérêt de cette troisième version ?

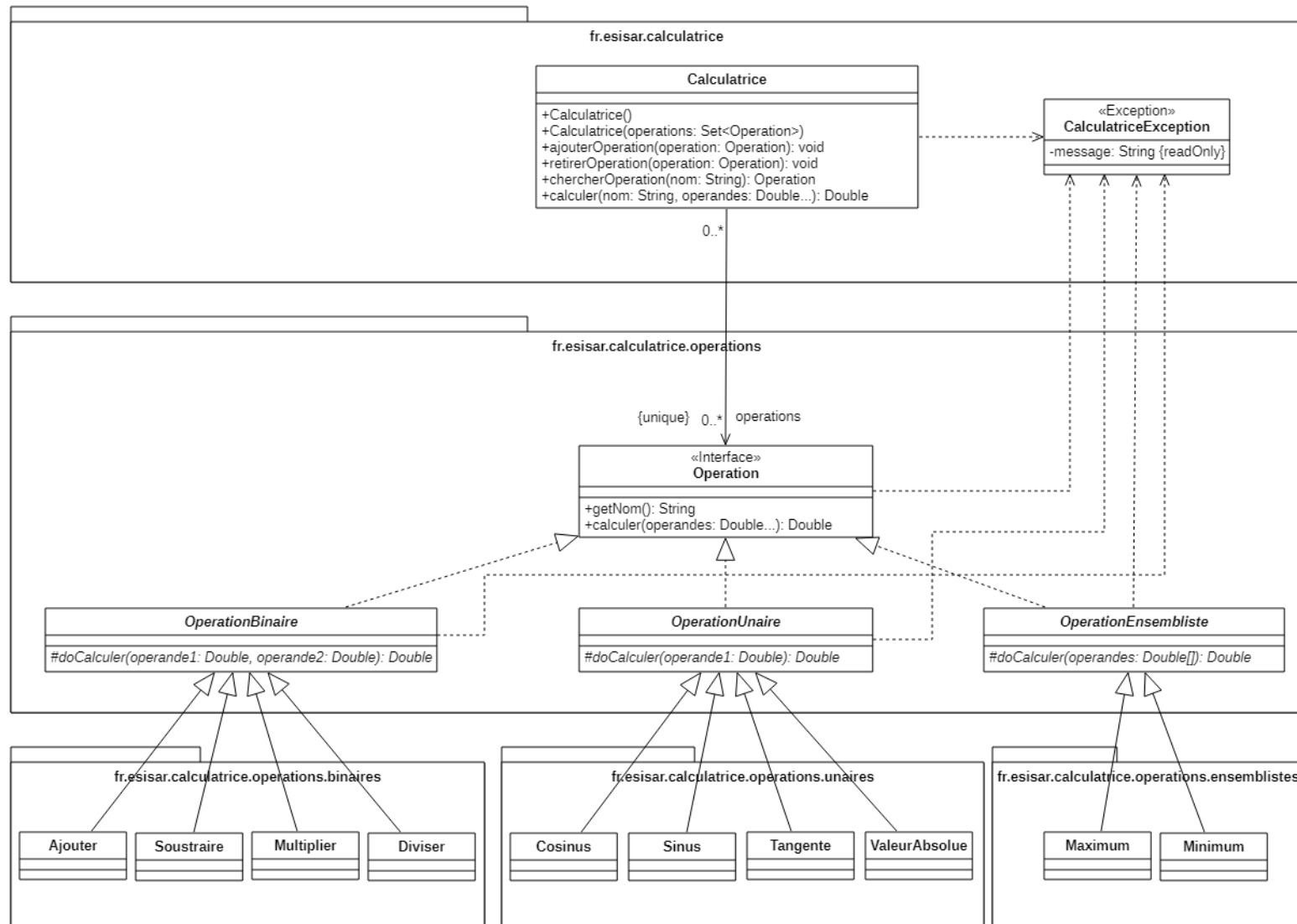


Figure 2 : Diagramme de classes de la version 3 de la calculatrice.