

Calculatrice – version 2¹

Compétences visées :

- Maîtrise de la configuration de son environnement de travail (système d'exploitation, versions de Java et Eclipse).
- Maîtrise du processus de développement logiciel d'une application développée en Java avec l'outil d'automatisation des tâches Apache Ant.
- 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-v2.
- 1.2. Dans votre projet Calculatrice-v2, vous allez développer une calculatrice qui supporte un nombre extensible d'opérations. Le diagramme de classes de cette nouvelle version de la calculatrice est donné dans la Figure 1.

A la différence de la version 1, vous devez ajouter la gestion des exceptions avec l'écriture d'une classe `CalculatriceException`, qui hérite de la classe standard `Exception` de Java. De plus, vous allez créer une classe par opération possible.

Le nom retourné par la méthode `getNom()` pour les classes `Ajouter`, `Soustraire`, `Multiplier` et `Diviser` est respectivement « + », « - », « * » et « / ».

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 (par exemple : l'une avec toutes les opérations – Ajouter, Soustraire, Multiplier et Diviser –, l'autre avec seulement quelques opérations).

Avant de commencer à développer, répondez à l'ensemble de ces questions :

- 1.2.1. Dans quel ordre allez-vous créer l'ensemble des éléments dans votre projet Eclipse ?
- 1.2.2. A quoi sert l'interface `Opération` ?
- 1.2.3. Dans la classe `CalculatriceException` (à titre exceptionnel, le constructeur et la méthode `toString` ont été représentés), à quoi correspond l'annotation `{readOnly}` à côté de l'attribut `message` ? Comment faut-il l'implanter ?
- 1.2.4. Quel est le fonctionnement de la classe `Calculatrice` ? A-t-elle des attributs ? Que représente la flèche entre la classe `Calculatrice` et l'interface `Operation` ? A quoi correspond l'indication `{unique}` ? Quelle structure de données devez-vous utiliser ? Quel(s) constructeur(s) faut-il implanter ?

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

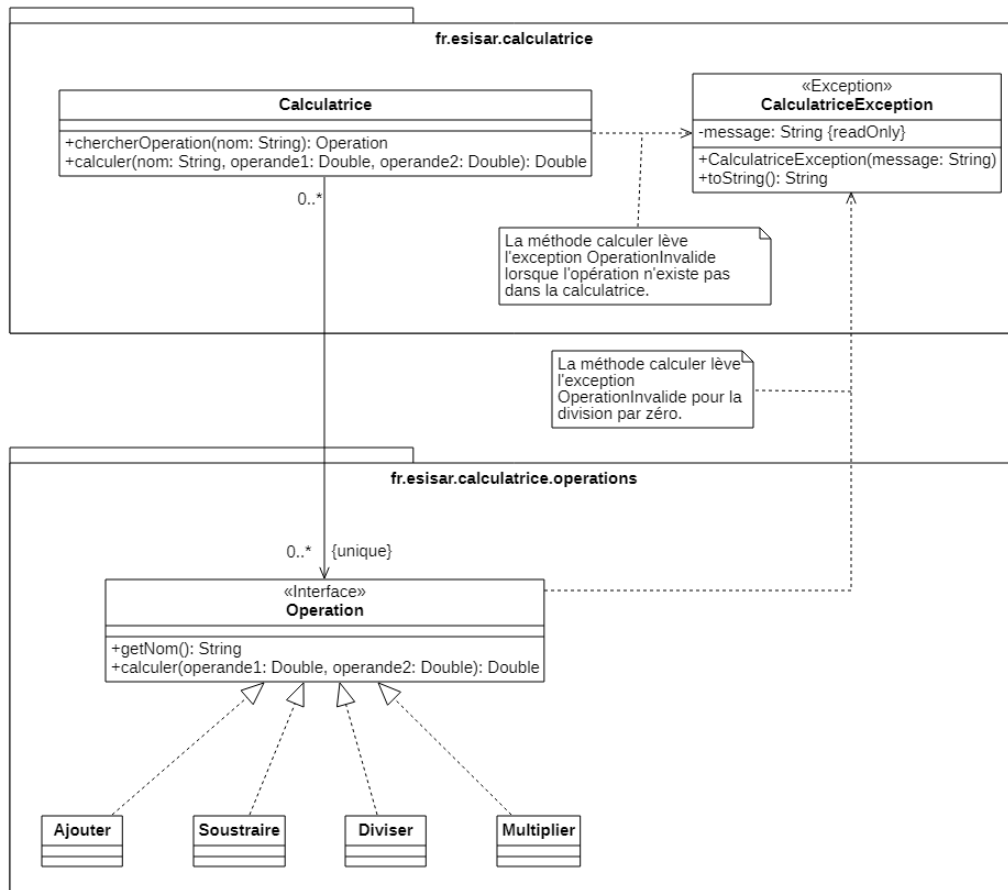


Figure 1 : Diagramme de classes de la version 2 de la calculatrice.

- 1.3. Ecrivez un fichier *build.xml* (à la racine du projet) afin d'automatiser, notamment, les tâches de compilation, de packaging et d'exécution de votre programme. L'arborescence de votre projet doit correspondre à celle donnée dans la Figure 2. Le répertoire *build/* (à la racine du projet) contient deux sous-répertoires : *classes/* pour les classes compilées et *jar/* pour l'archive exécutable du projet.

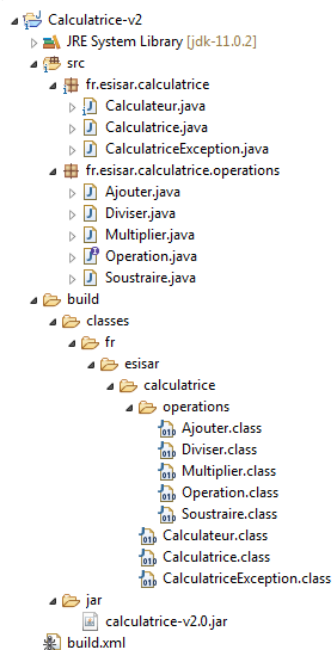


Figure 2 : Arborescence complète du projet.