# Rapport - Projet SDC

## Architecture générale de l’application

Nous avons découpé l’application en différents morceaux pour mieux nous y retrouvés. On retrouve :

* Les Exceptions : Ces exceptions héritent de ProcessingException qui permet l’affichage d’un message à l’utilisateur lors d’un mauvais fonctionnement sans pour autant interrompre le programme. Nous n’avons rajouté qu’une exception ‘VariableException’ lors des erreurs concernant les variables.
* Les Operations : Le package opérations englobe tous les traitements que l’on peut exécuter sur les objets héritent de Value. Elles font effet sur des éléments de la Stack<Value>. On peut donc ici voir les classes ‘SuperiorOperation’ ‘AddOperation’. Elles héritent toutes de soit ‘UnaryOperation’ soit ‘BinaryOperation’ qui définit leurs modes d’exécutions.
* Le package sdc : Il regroupe les classes SDC, SDCTextUI, Factory et Variable. Ce sont des classes isolées comparativement aux autres car elles n’héritent de rien. Cependant leurs rôles est primordial.
* Les Symbols : Les classes de ce package implémentes toutes la classe Symbol. Elles permettent de détecter certains mots-clefs et de déclencher une action en réponse. Nous nous en servons par exemple pour gérer les IF ELSE ENDIF. Lors de la détection d’un de ces mots, la classe correspondante déclenche les actions nécessaires au cas présent.
* Les Values : La classe Value implémente elle aussi Symbol car nous avons besoin de pouvoir détecter certains types de valeurs. Cette classe aurait très bien pu se retrouver dans le package Symbol mais de par son importance nous avons préfères lui faire une place à part. Ainsi ‘BooleanValue’, ‘NumericalValue’ ainsi que ‘AffectValue’ héritent de Value et au-dessous de ‘NumericalValue’ se trouve ‘RationnalValue’ et ‘IntegerValue’. Le fait que toutes ces classes soient regroupés sous la bannière de ‘Value’ nous permet de stocker tous ces types de données dans la même pile.
* Les Tests : Il y a aussi trois classes de test portant sur ‘BooleanValue’, ‘RationnalValue’ et ‘IntegerValue’.

## Modifications apportés

Nous avons globalement repris le même type d’architecture que le projet initial en ajoutant les classes :

Nous avons aussi massivement modifié la classe SDC non pas au niveau du fonctionnement global mais avec l’ajout de fonctions et surtout de condition d’exécution tout au long du processus.