

TD N°3 Ecrire une calculette JFlex/CUP

Reprendre le précédent TD ou créer un nouveau projet *Calculette* sur la base de l'archive <http://www.labri.fr/perso/clement/enseignements/compilation/calcullette2.tar.gz>.

Questions

1. La classe `java_cup.runtime.Symbol` contient les variables d'instances suivantes :

- `int left`
- `int right`
- `int sym`
- `java.lang.Object value`

Une instance de la classe `Symbol` correspond à un terminal de la grammaire. `left` et `right` sont réservées pour l'utilisateur. Il est d'usage d'y mettre les numéros de ligne et colonne du symbole. `sym` correspond à l'entier qui encode le terminal. `value` permet d'associer une valeur à un symbole.

Faire une version de débogage où chaque symbole est affiché en sortie erreur avec le numéro de ligne et le numéro de colonne correspondant.

2. La classe `java_cup.runtime.lr_parser` contient la méthode `void syntax_error(Symbol cur_token)` qui appelle par défaut `report_error("Syntax error", null)` ;
Surcharger la méthode `syntax_error` pour que le message d'erreur soit plus précis avec le numéro de ligne et le type d'erreur rencontré.
3. La méthode `public void unrecovered_syntax_error(Symbol cur_token)` est appelée quand une erreur de syntaxe ne permet plus de poursuivre l'analyse. Cette méthode appelle `report_fatal_error("Couldn't repair and continue parse", null)` ; qui met fin à l'analyse. Surcharger cette méthode pour arrêter correctement l'analyse après un affichage personnalisé du message.
4. La calculette fait un certain nombre de calculs et les affiche à la suite. Si l'un de ces calculs contient une erreur de syntaxe, tous les autres sont ignorés et l'analyseur syntaxique s'arrête. Utiliser le mot clef `error` pour provoquer une récupération après erreur et afficher `"ERROR"` pour l'expression erronée sans arrêter les calculs.