# Appréciations

Pour donner suite à votre utilisation de JavaCC et des AST d’Eclipse, donnez votre avis sur les deux implémentations en fonction de critères tels que : facilité d’implémentation, avantages et inconvénients, impact d’une modification du langage Java sur ce que vous avez réalisé, etc. (maximum une page).

## Miriam Davydov

**Avantages :** personnalisation

**Inconvénients :** répétition, difficile à maintenir, nouvel apprentissage

En mon sens, l’avantage primaire de JavaCC et AST c’est la personnalisation d’un *parser* qui répond mieux à nos besoins, mais tout en étant un avantage, celui-ci se transforme en inconvénient quand on conçoit que nos besoins ne sont pas si différents des autres programmeurs. Leur utilisation implique de recréer la roue et d’apprendre une syntaxe additionnelle rendant son usage compliqué.

Ces outils sont connus pour être difficile à maintenir aujourd’hui remplacé par des plugins maintenus par une équipe ou une communauté.

## Maxime Desmarais

Pour ma part, j’ai trouvé JavaCC très lourd à utiliser pour une grammaire qui fonctionne pour n’importe quel programme valide Java. Quoi qu’il puisse s’avérer très utile pour façonner de petits langages personnalisés, la taille finale du projet est un indice de sa complexité pour Java. Pour ce qui est des AST, je trouve que c’est une idée très intéressante mais qui nécessiterait un peu de peaufinage quant à son implémentation, la majorité des problèmes provenant de son initialisation. Cependant, une fois partis, AST s’avère beaucoup plus comprenable et facile d’utilisation comparé à JavaCC, pour des programmes Java évidemment.