

TP 4 : Génération de code (1)

Nous allons implémenter les **actions sémantiques** dans le même projet «**mips1Proj**». Pour cela :

- 1- Définir le type enum **Mnemonic** dans le package **net.mips.interpreter** et y déclarer les instructions P-code.
- 2- Définir la classe **Instruction** dans le package **net.mips.interpreter** :
 - Déclarer les attributs **mne** de type **Mnemonic** et **suite** de type **int**.
 - Générer les méthodes d'accès get et set.
 - Définir deux constructeurs avec et sans paramètres.
- 3- Modifier la classe **Symboles** :
 - Définir l'attribut **adresse**.
 - Générer les méthodes d'accès get et set.
- 4- Modifier la classe **ScannerWS** :
 - Définir l'attribut **offset**.
 - Générer les méthodes d'accès get et set.
 - Initialiser **offset** à -1 dans le constructeur.
 - Modifier la méthode **entrerSymb** afin d'effectuer la réservation de mémoire pour les constantes et les variables.
- 5- Modifier la classe **ParserWS** :
 - Définir les attributs **pcode** de type tableau dynamique d'instructions et **fluxCible** de type **PrintWriter**.
 - Générer les méthodes d'accès get et set.
 - Initialiser **pcode** dans le constructeur.
 - Définir les méthodes **generer1** et **generer2**.
 - Définir la méthode **savePcode** qui enregistre les instructions Pcode générées dans un fichier résultat (utiliser **fluxCible**).
 - Définir les **actions sémantiques** pour les règles :
 - program,
 - consts,
 - vars,
 - block,
 - affec,
 - fact,
 - term,
 - expr,
 - ecrire,
 - lire.