

TP 5 : Interpréteur Pcode

Nous allons implémenter l'**interpréteur Pcode** dans le même projet «**mips1Proj**». On considère pour cela la grammaire du langage Pcode :

```
PCODE      ::= INT NUM {INST_PCODE} HLT
INST_PCODE ::= ADD | SUB|EQL|...| [LDA | BZE|BRN|LDI] NUM
NUM        ::= CHIFFRE {CHIFFRE}
CHIFFRE    ::= 0|..|9
```

Les étapes à suivre sont :

- 1- Modifier le type enum **Mnemonique** du package **net.mips.interpreter** et y déclarer NUM et EOF.
- 2- Définir la classe **Symboles** dans le package **net.mips.interpreter** :
 - Déclarer les attributs **nom** de type **String** et **token** de type **Mnemonique**.
 - Générer les méthodes d'accès get et set.
 - Définir deux constructeurs avec et sans paramètres.
- 3- Définir la classe **ScannerPcode** dans le package **net.mips.interpreter** qui implémente l'analyseur lexical du langage Pcode. Cette classe ressemble à la classe Scanner. Elle définit les méthodes :
 - **lireCar()**.
 - **codageLex()**
 - **lireInst()**
 - **lireNombre()**
 - **symbSuiv()**
 - ...
- 4- Définir la classe **ParserPcode** dans le package **net.mips.interpreter** qui implémente l'analyseur syntaxique du langage Pcode. Cette classe ressemble à la classe Parser. Elle définit les méthodes :
 - **pcode()**
 - **instPcode()**
 - ...
- 5- Définir la classe **InterpreterPcode** dans le package **net.mips.interpreter** qui implémente l'interpréteur du langage Pcode. Cette classe définit les attributs et les méthodes suivantes :
 - tableau **pcode**
 - tableau **pile**
 - **pc**
 - **offset**
 - **sp**
 - **loadMnemonic(Mnemonique mne)**
 - **loadMnemonic(Mnemonique mne, int suite)**
 - **interInst()**
 - **interPcode()**