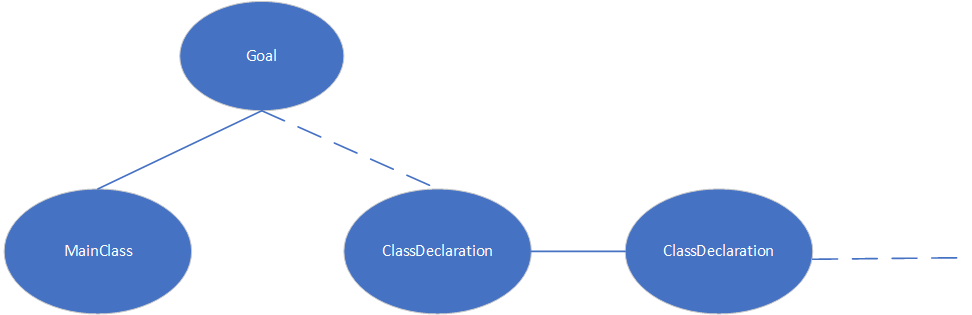
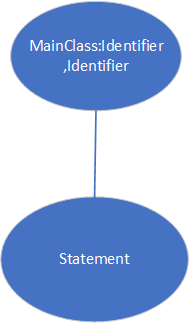
**MJava文法范式对应的抽象语法树结构**

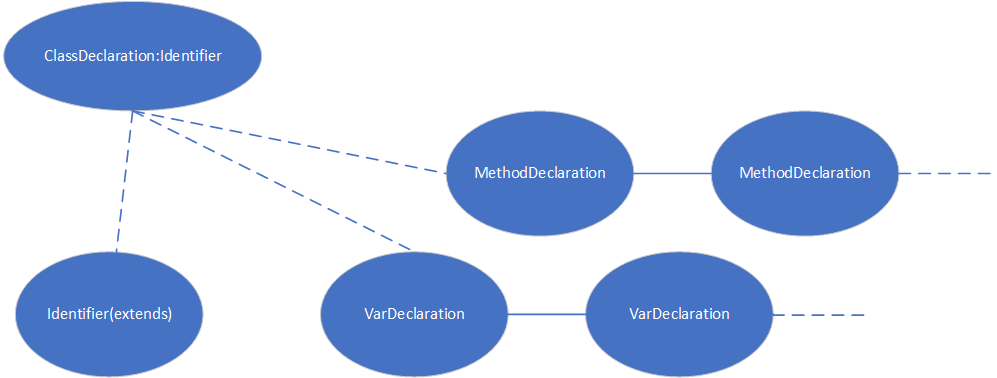
* Goal-> MainClass *{* ClassDeclaration *}* EOF

**

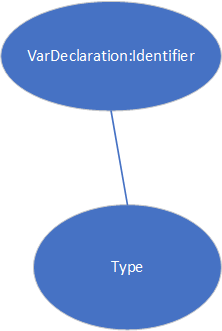
* MainClass->**"class"** Identifier **"{"** **"public"** **"static"** **"void"** **"main"** **"("** **"String"** **"["** **"]"** Identifier **")"** **"{"** Statement **"}"** **"}"**

**

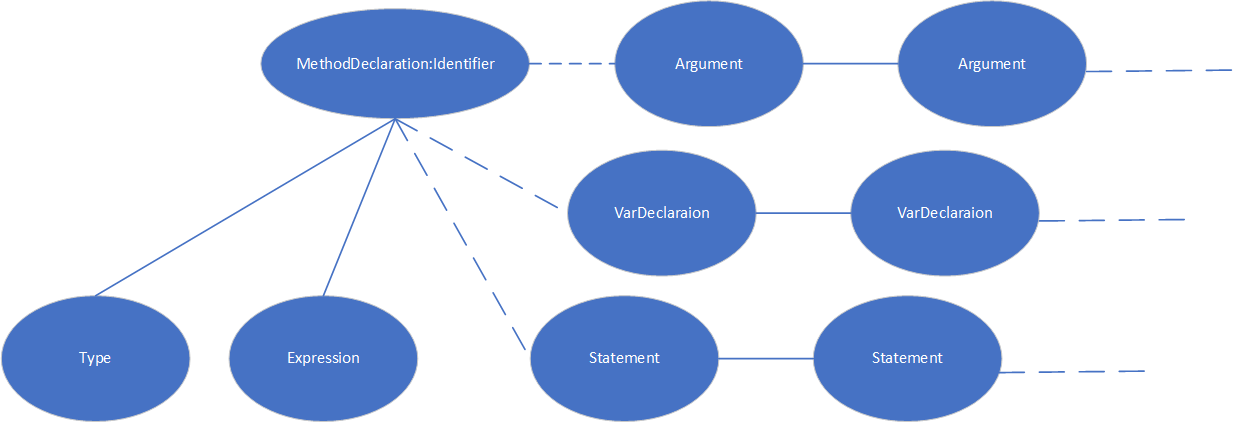
* ClassDeclaration->**"class"** Identifier *[* **"extends"** Identifier *]* **"{"** *{* VarDeclaration *}* *{* MethodDeclaration *}* **"}"**

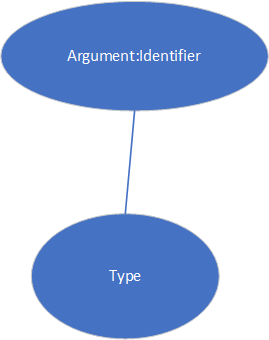


* VarDeclaration ->TypeIdentifier **";"**

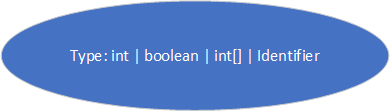
**

* MethodDeclaration->**"public"** Type Identifier **"("** *[* Type Identifier *{* **","** Type Identifier *}* *]* **")"** **"{"** *{* VarDeclaration *}* *{* Statement *}* **"return"** Expression **";"** **"}"**

****

****

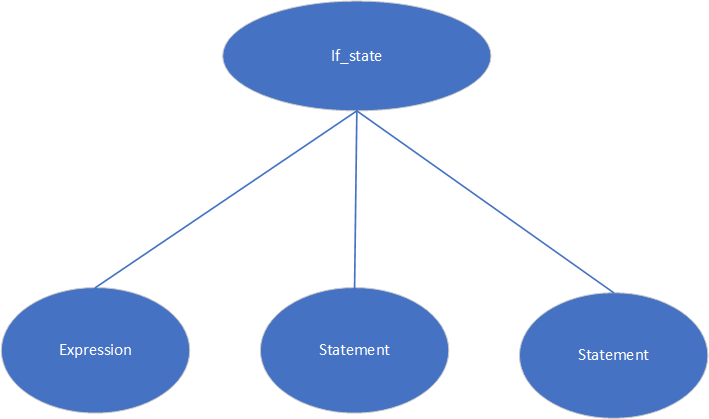
* Type->**"int"** **"["** **"]"***|***"boolean"***|***"int"***|* Identifier



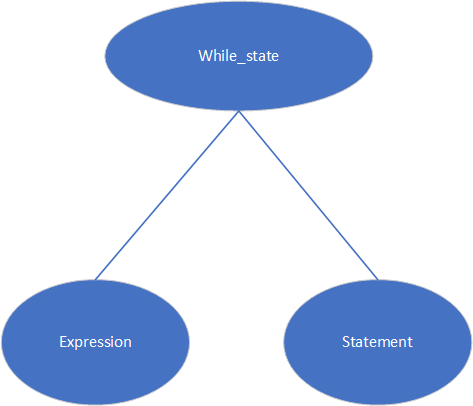
* Statement->**"{"** *{* Statement *}* **"}"**

****

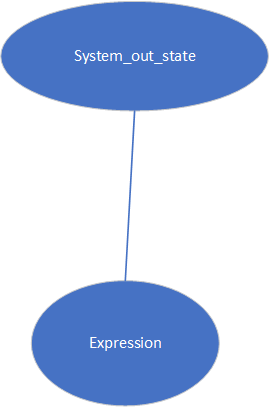
* Statement->**"if"** **"("** Expression **")"** Statement **"else"** Statement

**

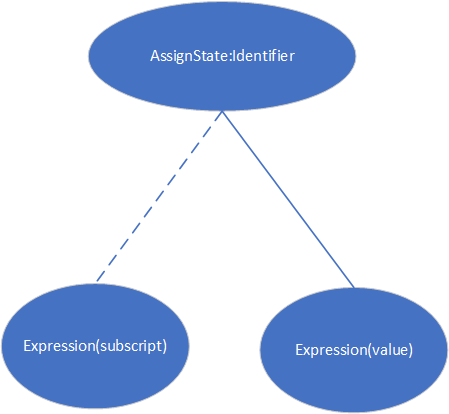
* Statement->**"while"** **"("** Expression **")"** Statement

****

* Statement->**"System.out.println"** **"("** Expression **")"** **";"**

****

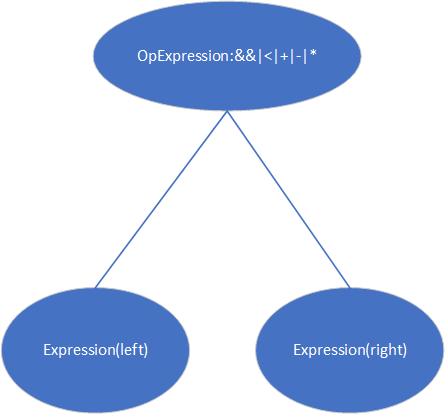
* Statement-> Identifier **"="** Expression **";"**
* Statement-> Identifier **"["** Expression **"]"** **"="** Expression **";"**

****

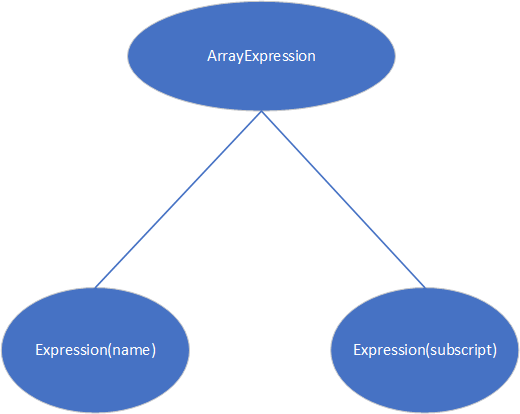
* Expression-> Expression **"["** Expression **"]"**

****

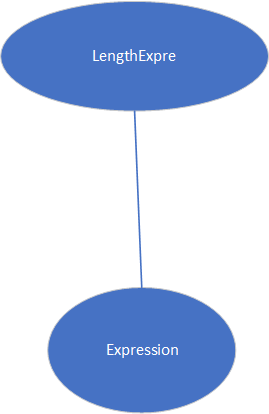
* Expression-> Expression *(* "&&" | "<" | "+" | "-" | "\*" *)* Expression



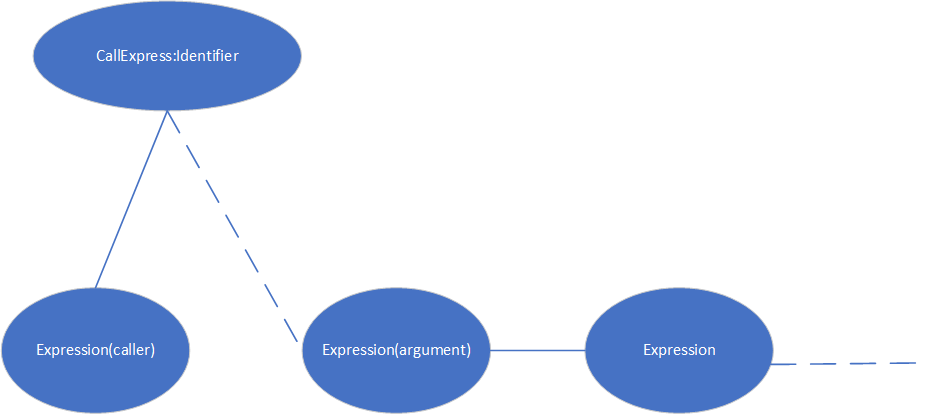
* Expression-> Expression **"["** Expression **"]"**



* Expression-> Expression **"."** **"length"**

**

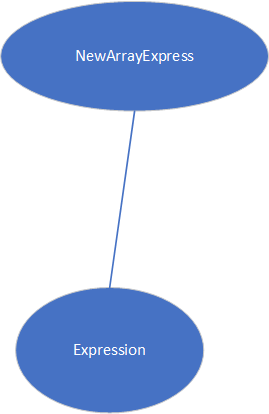
* Expression-> Expression **"."** Identifier **"("** *[* Expression *{* **","** Expression *}* *]* **")"**

**

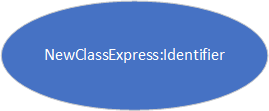
* Expression-> IntegerLiteral *|***"true"***|***"false"***|* Identifier *|***"this"**

**

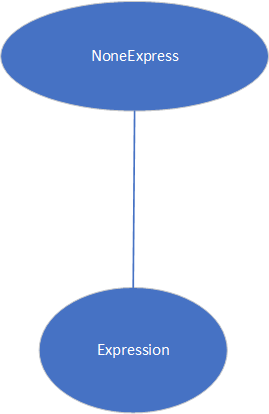
* Expression->**"new"** **"int"** **"["** Expression **"]"**

**

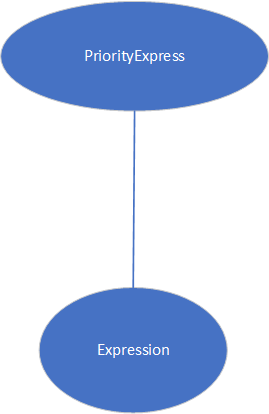
* Expression->**"new"** Identifier "(" ")"

**

* Expression-> **"!"** Expression

**

* Expression->**"("** Expression **")"**

**