```
grammar dk.sdu.mmmi.mdsd.Math with org.eclipse.xtext.common.Terminals

generate math "http://www.sdu.dk/mmmi/mdsd/Math"

Program:
    programName=ProgramName externals+=External* variableAssignments+=VariableAssignment+
;

ProgramName returns ProgramName:
    'program' name=ID
;

External:
    {External} 'external' name=ID '(' parameters+=ID? (',' parameters+=ID)* ')'
;

ExternalUse returns Expression:
    {ExternalUse} ref=[External] '(' exp+=Exp? (',' exp+=Exp)* ')'
;

//Variables:
//  variableAssignments+=VariableAssignment+
//;

VariableAssignment returns Variable: //Serves as a basis to retain results and to be the basis for lines in the dsl
    {VariableAssignment} 'var' name=ID '=' exp=Exp
;

Exp returns Expression: //Addition and subtraction - Can boil down to MultDiv
    MultDiv (('+' {Plus.left=current}| '-' {Minus.left=current}) right=MultDiv)*
;

MultDiv returns Expression: //Multiplication and devision - Can boil down to MultDiv
    Primary (('*' {Multiplication.left=current}| '/' {Division.left=current}) right=Primary)*
;

Primary returns Expression: //Numbers and things that should be computed down to numbers before use
    Number | Parenthesis | VariableUse | LocalAssignment | ExternalUse
;

Parenthesis returns Expression: //Serves to support the use of parentheses as
```

```
  a base
42    {Parenthesis} '(' exp=Exp ')'
43 ;
44
45 Number returns Expression: //A basic number
46    {ExplicitNumber} value=INT
47 ;
48
49 VariableUse: //Using a previously defined variable
50    {VarUse} ref=[Variable]
51 ;
52
53 Assignment returns Variable:
54    {Assignment} name=ID '=' exp=Exp
55 ;
56
57 LocalAssignment: //This is kind of like a using statement, where an alias is
   made for an expression or simmilar that only exists in the body of the let
   statement
58    {Local} 'let' assignment=Assignment 'in' exp=Exp 'end'
59 ;
60
61
```