

## Compiladores

### Laboratorio 18

#### Objetivo

Implementar la generación de código para una gramática con funciones.

#### Programa

- `Program ::= VarDecList FunDecList`
- `VarDecList ::= (VarDec)*`
- `FunDecList ::= (FunDec)+`
- `FunDec ::= fun Type id ([ParamDecList] ) Body endfun`
- `Body ::= VarDecList StmtList`
- `ParamDecList ::= Type id (, Type id)*`
- `VarDec ::= var Type VarList ;`
- `Type ::= id`
- `VarList ::= id (,id)*`
- `StmtList ::= Stmt ( ; Stmt )*`
- `Stmt ::= id = CExp |`
  - `print ( CExp )`
  - `if CExp then Body [else Body] endif`
  - `while CExp do Body endwhile`
  - `return ( [CExp] )`
- `CExp ::= Exp [( < ) Exp]`
- `Exp ::= Term (( + | - ) Term)*`
- `Term ::= Factor (( * | / ) Factor)*`
- `Factor ::= id | Num | Bool | ( Exp ) | ifexp ( CExp , CExp , CExp ) | id ( [ArgList] )`
- `ArgList ::= CExp (, CExp)*`
- `Bool ::= true | false`

#### Problema

Completar los `GenCodeVisitor` para la clase `FunDec*`, `ReturnStatement*` y `FCallExp*`.