Compiladores
Laboratorio
10
Visitors II

Programa

Se tiene implementado la siguiente gramática,

- Program ::= StmtList
- StmtList := Stmt ('**;**' Stmt)*
- Stmt ::= **id** '**=**' Exp | '**print**' '**(**'Exp '**)**'
- Exp ::= Term ((´**+**´ | ´**-**´) Term)*
- Term ::= Factor ((´***´|´**/**´) Factor)*
- Factor ::= **id** | **Num** | ´**(**´ Exp ´**)**´

```
g++ main.cpp exp.cpp parser.cpp scanner.cpp token.cpp visitor.cpp
```

Problema 1

- Program ::= StmtList
- StmtList := Stmt ('**;**' Stmt)*
- Stmt ::= **id** '**=**' CExp | '**print**' '**(**' CExp '**)**' | "**if**" CExp "**then**" StmtList ["**else**" StmtList] "**endif**"
- CExp ::= Exp [("<"|"<="|"==") Exp]
- Exp ::= Term ((´**+**´ | ´**-**´) Term)*
- Term ::= Factor ((´***´|´**/**´) Factor)*
- Factor ::= **id** | **Num** | ´**(**´ CExp ´**)**´

Ejemplos

```
x=4;
y=5;
if x<y then print(x) else print(y) endif;
print(10)

Visitor imprimir
x=4
y=5
if x<y then
      print(x)
else
      print(y)
endif

Visitor ejecutar
5
10
```

Instrucciones:

1. Agregar el token (MENOR, IF, ELSE, ENDIF, etc.).
2. Modificar el scanner. Revisar que la lista de tokens reconocidos sea correcta.
3. Las operaciones de relación son operaciones binarias (que devuelven 0 o 1), no necesitas una nueva clase.
4. El **if** necesita una clase **IFStatement** tienes que modificar **exp.h** y **exp.cpp**:
    - CExp*
    - list<Stm*> slist1
    - list<Stm*> slist2
    - Constructor, destructor y su acepta.
5. Modificar el visitor y dejar las funciones vacías.
6. Crear una función en parser.h y parser.cpp
    - Exp* parseCExp ()
    - No olvidar modificar parseStatement que deberían llamar parseCExp.
7. Modificar parseStatement para que reconozca el if.

    - Deberia crear el *Exp, slist1, slist2
    - Match(if)
    - parseCExp
    - Match(then)
    - parseStm()
    - while(match(COMA)) parseStm() , pusback(slist1)
    - Match(else)
        - parseStm()
        - while(match(COMA)) parseStm(), pusback(slist2)
    - Matcch(endif)
    - Devolver new **IFStatement**

8. Modificar el visitor:
    - Completar las funciones print:
        - visit(AssignStatement* stm)
        - visit(PrintStatement* stm)