Compiladores
Laboratorio 10

Objetivo
Elaborar el intérprete de una gramática con strings.

Programa
Se tiene implementado la siguiente gramática,

- Program ::= StmtList
- StmtList ::= Stmt **(;** Stmt **)***
- Stmt ::= **id =** CExp |
  print (CExp **)** |
  if CExp **then** StmtList [**else** StmtList] **endif** |
  while CExp **do** StmtList **endwhile**
- CExp ::= Exp [(**<**|**<=**|**==**) Exp]
- Exp ::= Term ((**+** | **-**) Term)*
- Term ::= Factor ((*****|**/**) Factor)*
- Factor ::= **id** | **Num** | **(** CExp **)**| **ifexp (**CExp **,** CExp **,** CExp **)**

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

Problema
Se solicita implementar:

- Program ::= StmtList
- StmtList ::= Stmt **(;** Stmt **)***
- Stmt ::= **id =** CExp |
  print (CExp **)** |
  if CExp **then** StmtList [**else** StmtList] **endif** |
  while CExp **do** StmtList **endwhile**
  for id in range(Cexp**,**Cexp**,**Cexp**) do** StmtList **endfor**
  for id in String **do** StmtList **endfor**
- CExp ::= Exp [(**<**|**<=**|**==**) Exp]
- Exp ::= Term ((**+** | **-**) Term)*
- Term ::= Factor ((*****|**/**) Factor)*
- Factor ::= **id** | **Num** | **(** CExp **)**| **ifexp (**CExp **,** CExp **,** CExp **)** | String
- String ::= **“[A-Za-z0-9]+” | id [** Cexp **]**

Ayuda:

```
class StringLiteral : public Exp {
public:
   string value;
   StringLiteral( const string& )
   int accept(Visitor* visitor) override;
   string acceptString(Visitor* visitor) override;
   ~StringLiteral()
};
```

```
class IndexExpr : public Exp {
public:
   string id;
   Exp* index;
   string literalValue;
   IndexExpr(const string& , Exp* );
   int accept(Visitor* visitor) override;
   string acceptString(Visitor* visitor) override;
   ~IndexExpr();
};
```

Entrada

```
x = "abc";
print(x + "d");
x = "repe";
print(x[0]+x[1])
print(x * 2);
for i in range(1, 6, 2) do print(i) endfor;
 z = “Hola”;
for i in z do print(i) endfor;
```

Salida

```
abcd
re
reperepe
1
3
5
H
o
l
a
```