

Prueba 4 -Evaluación continua 1

Los archivos prueba4.hh y prueba4.cpp definen e implementan la estructura y funcionalidad de los árboles que representan lo sentencias en el lenguaje imperativo dado por la siguiente gramática:

- $\text{Program} ::= \text{StmtList } \$$
- $\text{StmtList} ::= \text{Stmt } (';' | '\n')^* \text{Stmt}$
- $\text{Stmt} ::= \text{id } '=' \text{Exp} \mid \text{'print' } '(' \text{Exp } ')'$
- $\text{Exp} ::= \text{Term } (('+' | '-') \text{Term})^*$
- $\text{Term} ::= \text{Factor } (('*' | '/') \text{Factor})^*$
- $\text{FExp} ::= \text{Factor } ['* \text{FExp}]$
- $\text{Factor} ::= \text{id} \mid \text{num} \mid '(' \text{Exp } ')'$

Por ejemplo, si se ingresa el fichero:

ejemplo.txt

x=5; y=4 \n

z=20 \n

print(x+y); print(x+z); print(y+z)

El analizador nos proporciona:

program:

```
{  
x = 5;  
y = 4;  
z = 20;  
print(x + y);  
print(x + z);  
print(y + z);  
}
```

execute:

```
9  
25  
24
```

Ahora, se **solicita que extienda el programa y este pueda hacer uso de las funciones trigonométricas**, para esto se tiene la gramática:

- $\text{Program} ::= \text{StmtList } \$$
- $\text{StmtList} ::= \text{Stmt} ((';' \mid '\backslash n') \text{Stmt})^*$
- $\text{Stmt} ::= \text{id } '=' \text{Exp} \mid \text{'print' } '(' \text{Exp } ')'$
- $\text{Exp} ::= \text{Term} (('+' \mid '-') \text{Term})^*$
- $\text{Term} ::= \text{Factor} (('*' \mid '/') \text{Factor})^*$
- $\text{FExp} ::= \text{Factor} ['*' \text{FExp}]$
- $\text{Factor} ::= \text{id} \mid \text{num} \mid '(' \text{Exp } ') ' \mid \text{sin } '(' \text{Exp } ') ' \mid \text{cos } '(' \text{Exp } ') '$

Un ejemplo, del resultado que se espera:

ejemplo.txt

```
x=30 \n
y=180 \n
print(sin(x)); print(cos(y))
```

program:

```
{
x = 30;
y = 180;
print(sin(x));
print(cos(y));
}
```

execute:

```
0.5
-1
```