

Cuarto boletín de prácticas sobre MMPor

IG29: Compiladores e intérpretes

Sexta sesión de prácticas

(1) Adapta el diseño y la implementación de tu calculadora para conseguir que su comportamiento pase a ser el siguiente:

- Cada línea podrá ser una asignación o bien una sentencia de escritura por la salida estándar.
- Las asignaciones ya no supondrán impresión alguna.
- Las sentencias de escritura consistirán en una palabra reservada, `print`, seguida de una secuencia de una o más expresiones separadas por comas. La palabra reservada ha de escribirse íntegramente en minúsculas; `Print` o `PRINT`, por ejemplo, se interpretarían como identificadores.
- En realidad, en la secuencia de elementos a imprimir, en el lugar de las expresiones también podrán aparecer literales de cadena: secuencias delimitadas por caracteres doble comilla y sin tabuladores, saltos de línea ni dobles comillas en su interior. Obtener el valor de un literal de cadena consistirá en tomar el lexema sin sus comillas delimitadoras.
- Los valores de los elementos de una sentencia de escritura se imprimirán sin separación entre ellos y la ejecución de la sentencia finalizará con la impresión de un salto de línea.
- Los valores enteros se mostrarán en base diez, sin prefijos de ceros innecesarios y con un signo menos si procede.

Por supuesto, la opción `-s` debe seguir funcionando con normalidad, incluso con las nuevas clases de nodos que hayas podido introducir en tu diseño.

(2) Adapta tus pruebas a la nueva situación y verifica que tu calculadora las supera sin fallos.

(3) Entrega en un paquete `entrega05.tgz` los ficheros siguientes:

- `diseño.txt`.
- `mmpor.mc`, `AST.py`, `memo.py` y `errores.py`.
- Los correspondientes ficheros de prueba.

(4) Puedes ir preparándote para el trabajo del próximo boletín si piensas en cómo introducirías en tu calculadora una estructura de control condicional y qué utilizarías como condiciones:

- ¿Mejor una estructura sin cierre explícito, tipo `if/then/else`, o con él, `if/then/else/end`? En el primer caso, ¿necesitarías introducir el concepto de sentencia compuesta?
- ¿Permitirías que el `else` fuera opcional? ¿Podría esto dar lugar a ambigüedades?
- ¿Introducirías algún tipo de `elif` en la sintaxis de la estructura de control?
- ¿Mejor introducir un tipo lógico explícito con valores `true` y `false` o considerar que los valores de otros tipos tienen un valor de verdad asociado? En el segundo caso, ¿considerarías como “verdaderos” los enteros no nulos, los positivos o los no negativos?
- ¿Qué nuevos operadores introducirías para escribir cómodamente condiciones? ¿Qué comparaciones? ¿Introducirías también operaciones lógicas como `and`, `or` y `not`?