

Resolución de Problemas y Algoritmos

Clase 3: Write y Read

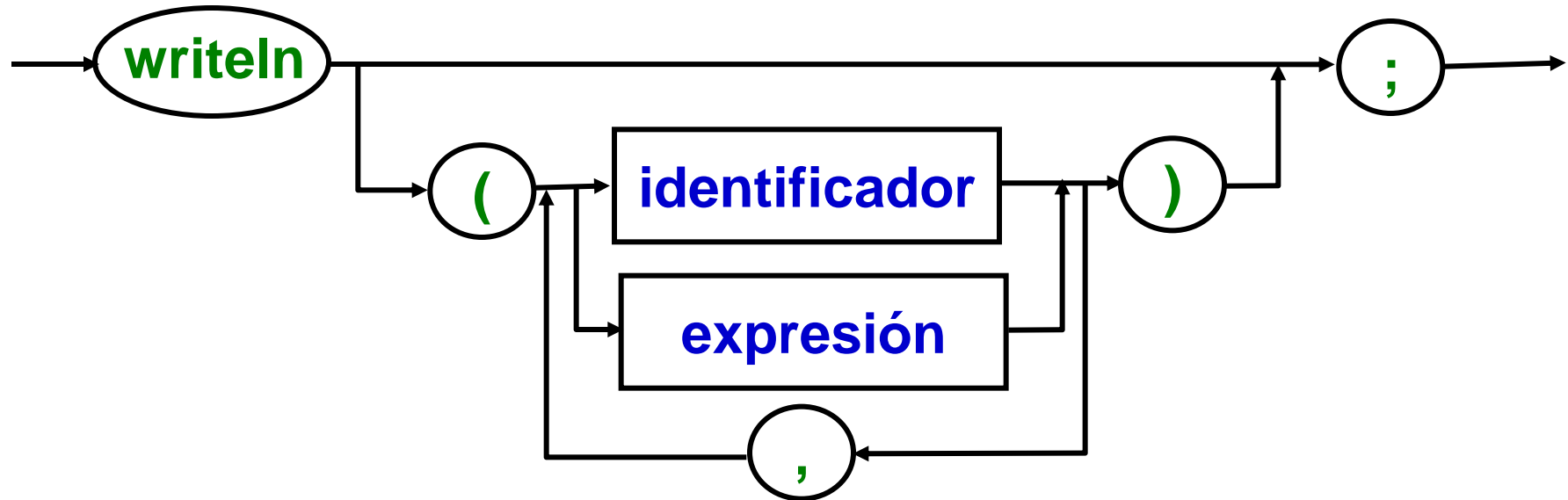


Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
Bahía Blanca - Argentina

Write y Read

Procedimiento WriteIn

Diagrama Sintáctico:

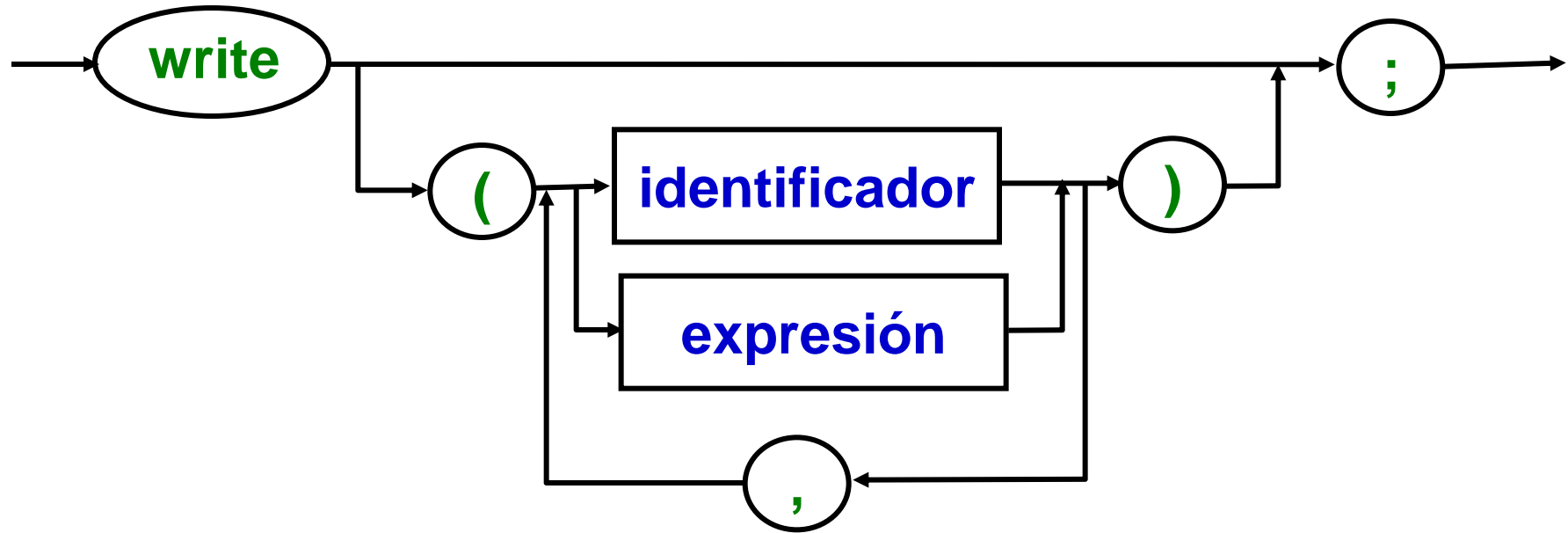


Muestra el contenido de cada uno de los parámetros (en la misma línea) y luego avanza a la próxima línea.

Ej: `writeIn('El tamaño de la puerta es de ' , Ancho, ' por ' , Alto);`
`writeIn;`
`writeIn('Que es igual a ' , Ancho * Alto, ' cm cuadrados ');`

Procedimiento Write

Diagrama Sintáctico:



Muestra el contenido de cada uno de los parámetros (en la misma línea).

Ej: `write('El tamaño de la puerta es de ', Ancho, ' por ', Alto, ' = ', Ancho * Alto);`

Write versus WriteIn

- Usualmente, se utiliza el **writeIn** para mostrar resultados. Sin embargo, si la cantidad de información es mucha es conveniente distribuirla en varios **write** para una **mayor claridad**.

writeIn(P1,P2,P3,P4,P5,P6,P7,P8);

es equivalente a

write(P1,P2,P3);

write(P4,P5,P6);

writeIn(P7,P8);

Write con formato

- **Generalmente, se necesita mostrar la información con un formato específico o en forma tabular.**
- **Se utilizará un ancho de campo (fw) para formatear la salida.**
- ***Tanto write como writeln aceptan formato.***

Formato:

write(<ParámetroEntero>:fw, <OtrosParam>);

write(<ParámetroReal>:fw:dp, <OtrosParam>);

Enteros con formato

- Para formatear un valor Entero se utilizará un **ancho de campo (fw)** que especifica la cantidad de dígitos a ser mostrado. Obs: el **signo (-)** ocupa 1 lugar.
- Si la cantidad de dígitos a mostrar **es igual** a fw
Se muestra el **valor sin modificación**.
Ej.: `write(123:3);` *Muestra: 123*
- Si la cantidad de dígitos a mostrar **supera** a fw
Se **agregan los lugares necesarios**.
Ej.: `write(123:1);` *Muestra: 123*
- Si la cantidad de dígitos a mostrar **es inferior** a fw
Se **agregan espacios, justificando el nro a la derecha**.
Ej.: `write(123:6);` *Muestra: ⇨⇨⇨⇨⇨123*

Reales con formato

- Para formatear un valor Real además del **ancho de campo (fw)** se utilizará un valor **cantidad de decimales (dp)** que especifica la cantidad de dígitos decimales a ser mostrado.
- El **ancho de campo (fw)** consistirá de la **cantidad deseada** de dígitos en la **parte entera**, más dígitos en la **parte decimal**, más el lugar del **punto decimal (.)**
*Ejemplo: Si X es tipo Real y contiene sólo valores en el rango -99.9 a 999.9 se puede utilizar **X:5:1***
- La parte decimal es redondeada, la parte entera mantiene la cantidad de dígitos (igual que enteros).

Reales con formato

Considere X un valor *Real* y el formato **:5:1**

Valor de X	Salida
-99.42	-99.4
0.123	⇒⇒0.1
-9.53	⇒-9.5

Valor de X	Salida
-25.55	-25.6
99.999	100.0
999.43	999.4

Read: comportamiento del buffer

- Al ejecutar un **READ**, el programa se suspende y todo lo ingresado por teclado se almacena en el **buffer de lectura**.
- Al presionar la tecla **ENTER** la ejecución continúa y el procedimiento **READ** lee del **buffer** los valores que necesita para asociar a las variables.
- De esta manera, si uno escribe, borra y vuelve a escribir el programa no tiene que deshacer acciones.
- Considere el programa ejemplo “eje_read”:
 - si ingresa **1 2 3** el programa calculará $D = 6$, correctamente.
 - Si ingresa **1 2 3 4 5**, el programa también calculará $D = 6$ correctamente!
 - Si ingresa **1 2.7 3**, el programa dará error en tiempo de ejecución (*Invalid Numeric Format*)