

Instalación

Para utilizar el compilador GNAT, primeramente, se deben seguir los pasos de instalación especificados por el instalador GNAT-3.15p-nt.exe (Descargar de la página/Descargas).

Instalar preferentemente en el directorio C:\GNAT.

Posteriormente, para poder ejecutar los ejemplos de ADA propuestos, será necesario que estos programas se encuentren en el directorio \GNAT\bin

Los programas ADA poseen extensión “.adb”. Para poder realizar la compilación y generación del programa ejecutable es necesario el siguiente comando, el cual se ingresa en una interfaz de comandos de MSDOS

gnatmake -gnatf <nombre de archivo.adb>

Cabe aclarar por simple conveniencia, que es necesario estar ubicado en el directorio donde se encuentra el archivo a compilar. Luego de ejecutar el comando antes mencionado, se genera un archivo ejecutable el cual permite ver los resultados del programa antes compilado.

1. PASAJE DE PARÁMETROS

La semántica de pasaje de parámetros en ADA se representa anteponiendo en la definición de los parámetros (formales) alguna de las siguientes especificaciones **IN**, **OUT** o **IN OUT**. Si se omite alguno de ellos, se toma por defecto **IN**.

La definición de cada forma de pasar los parámetros es la siguiente:

- **IN** : El parámetro es pasado por copia valor
- **OUT** : El parámetro es pasado por resultado
- **IN OUT** : El parámetro formal es pasado por copia valor resultado

Existen modalidades para pasar estos parámetros al momento de invocar a subprogramas. Estos modalidades: **NAMED** y **DEFAULT**.

Los parámetros **NAMED** expresan explícitamente el valor que será pasado al parámetro formal:

<Nombre del Parámetro Formal> => <Valor/Variable>

La modalidad **NAMED** permite también los parámetros por defecto. Esto significa que a veces uno o más parámetros definidos con **IN**, toman el mismo valor en cada invocación. Para ello se provee una expresión por defecto en la especificación del subprograma (parámetros formales) y se omite el parámetro real correspondiente en el llamado al subprograma.

Ejemplos

A continuación se detalla una breve explicación de los ejemplos que se incluyen en el material de práctica.

Ejemplo 1: ADA_Ej1.adb.

Parámetros implicados IN, OUT, IN OUT.

Se ingresan dos números. Al primero de ellos, se le suma dos unidades y se lo multiplica por el segundo (procedimiento PROD).

Finalmente, se muestra el resultado de los parámetros reales luego de invocar a PROD.

Ejemplo 2: ADA_Ej2.adb.

Parámetros implicados IN OUT.

Se tiene un arreglo de N elementos, donde cada celda del arreglo tiene el valor 1.

Se muestra como influye la variación del índice de recorrido del arreglo (por el modo IN OUT) dentro del procedimiento SILLY.

Finalmente, se muestra el valor de tal índice y el estado del arreglo luego de invocar a SILLY.

Ejemplo 3: ADA_Ej3.adb.

Parámetros implicados NAMED PARAMETERS.

Se tiene un procedimiento (CUADRATIC) que calcula las raíces de una ecuación cuadrática ($Ax^2+Bx+C=0$).

Se pasan como parámetro los términos A, B y C de dicha expresión, para luego mostrar el valor de las raíces encontradas.

Ejemplo 4: ADA_Ej4.adb.

Parámetros implicados DEFAULT PARAMETERS.

Se tiene un procedimiento (GENERIC) el cual establece valores por defecto a sus parámetros.

Luego de diferentes invocaciones a dicho procedimiento, se muestran los resultados obtenidos sobre tales parámetros.

2. CONVERSIONES EXPLICITAS

Ejemplo 1 : Conv_ej1.adb

En la primera sentencia se convierte una expresión real a una entero y se le asigna a un entero. En la segunda, se convierte un entero a un real.

Ejemplo 2 : Conv_ej2.adb

Ejemplo sencillo de un procedimiento en el que dado dos valores enteros obtenidos desde teclado se los suma e imprime el resultado. Utiliza type

Ejemplo 3 : Conv_ej3.adb

Ejemplo simple de conversiones explícitas con tipos derivados. Utiliza Type is new

3. EJERCICIOS PROPUESTOS

PASAJE DE PARAMETROS

EJERCICIO 1:

- Escribir una función ADA llamada `ADD` la cual retorne la suma de dos enteros pasados como parámetro.
- Dicha función debe tomar el valor por defecto 1 para el segundo parámetro.
- Construir la de tal forma que sea posible intercambiar los valores de los parámetros reales al momento de invocarla, pero cada uno de los llamados deben ser equivalentes entre sí.

EJERCICIO 2:

- Reescribir la especificación del procedimiento `GENERIC`s del ejemplo `ADA_Ej4.adb`.
- Dicho cambio debe reflejar que el parámetro `PARAM_1` es igual a "Valor_A" si `PARAM_2` es igual a "Valor_C". Caso contrario, debe valer "Valor_Z".
- Construya dicho procedimiento de forma tal que permita visualizar cuando es evaluado cada parámetro.

EJERCICIO 3:

- Construir un programa ADA para cada uno de los siguientes casos, modificando el ejemplo mencionado para ver que efectos produce:
- Sobre `ADA_Ej1.adb`: Hacer un procedimiento tal que posea un parámetro pasado por el modo `IN` y al mismo se le asigne el valor de otro parámetro pasado por `OUT`. Explique el resultado.
- Sobre `ADA_Ej1.adb`: Realice un procedimiento en el cual un parámetro `C` pasado por el modo `OUT` se le sume una constante. Indicar que ocurre. Que pasaría si se especifica la siguiente instrucción: `IF C=3 THEN?`