

Montes Barreras | Zamarripa Moreno

ESPECIFICACIÓN LENGUAJE DE PROGRAMACIÓN

Universidad Autónoma de Sinaloa

Facultad de Ingeniería
Software de Sistemas

M.C. Edgar Omar Pérez Contretas

Montes Barreras Luis Geronimo

Zamarripa Moreno José Alberto

Grupo 401

Viernes, 16 de Octubre del 2020

Contenido

1. ESPECIFICACIÓN DEL CUERPO DEL PROGRAMA.....	3
a. SECCION DE DECLARACION DE VARIABLES	3
b. DECLARACION DE CONSTANTES	4
c. CUERPO PRINCIPAL DEL PROGRAMA	4
3. ESPECIFICACIÓN PARA DECLARAR VARIABLES	5
4. ESPECIFICACIÓN PARA DECLARAR CONSTANTES.....	7
5. ESTRUCTURA CONDICIONAL SIMPLE (IF...ELSE).....	9
6. ESTRUCTURA DE CONTROL (CICLO FOR)	11
7. ESPECIFICACIÓN PARA IMPRIMIR EN PANTALLA Y LEER VARIABLES DESDE EL TECLADO.....	13
a. IMPRIMIR EN PANTALLA	13
b. LEER VARIABLES DESDE TECLADO.....	13
8. PALABRAS RESERVADAS	14
9. OPERADORES ESPECIALES	15

1. ESPECIFICACIÓN DEL CUERPO DEL PROGRAMA

El lenguaje de programación ****Nombre**** tendrá el cuerpo que se muestra en la imagen, contando con un total de tres secciones: declaración de variables, declaración de constantes y cuerpo principal del programa que es donde se encontraran las instrucciones a ejecutar.

```
####Declaración de variables####
var
    entero : variable1;

####Declaración de constantes####
const
    entero : const1 := 10;

####Cuerpo principal del programa####
inicio
    ####Instrucciones####
    variable1 := const1 + 1;
fin
```

Ilustración 1 Cuerpo del Programa

a. SECCION DE DECLARACION DE VARIABLES

La sección de declaración de variables debe aparecer explícitamente al inicio del documento, si la palabra reservada *'var'* aparece en otra parte del programa se marcará un error y la compilación no se llevará a cabo. La forma correcta para declarar variables es colocar la palabra reservada *'var'* al inicio del documento, de esta manera estamos abriendo el campo de declaración de variables; después se realiza la declaración de las variables (esto se verá en una sección mas adelante). Se podría pensar que dejar el campo var sin declaraciones marcaria un error, pero este no es un error, por lo tanto, la compilación continua normalmente. Cabe resaltar que es obligatorio que la palabra reservada *'var'* aparezca en el documento, de otro modo el compilador marcara error.

b. DECLARACION DE CONSTANTES

La sección de declaración de constantes funciona de manera similar a la sección de declaración de variables, a diferencia de que esta sección debe aparecer después de la palabra reservada *'var'*, de otra forma marcará error. Al igual que la palabra reservada *'var'*, es obligatorio que la palabra reservada *'const'* aparezca en el documento, de otra forma el compilador marcará error. No se detalla todo sobre esta sección dado que se abordará en una sección más adelante.

Una similitud entre estas dos secciones radica en la exclusividad que tiene, dado que las instrucciones de declaración de variables son únicas y exclusivas para este campo, de igual manera para la declaración de constantes, es decir, que si se encuentra una declaración de una variable o una constante fuera de su campo de declaración el compilador marcará un error.

La manera correcta de declarar variables y constantes se abordará en secciones más adelante.

c. CUERPO PRINCIPAL DEL PROGRAMA

Las palabras reservadas *'inicio'*, *'fin'* marcan el inicio y el final del cuerpo principal del programa respectivamente; dentro de estas palabras se deben encontrar todas las instrucciones que el programa debe ejecutar. Si una instrucción es colocada antes de *'begin'* o después de *'end'* el compilador marcará error. El orden de aparición estas dos palabras reservadas es:

1. *'inicio'* -> Indicamos el inicio de la sección de instrucciones.
2. *'fin'* -> Indicamos el final de la sección de instrucciones.

Solo deben aparecer una vez en todo el documento, de lo contrario el compilador marcará error.

Las instrucciones contenidas entre las palabras reservadas *'inicio'* y *'fin'* son instrucciones que se deben compilar y ejecutar. De manera general se establece que todas las instrucciones terminan con punto y coma (;) para indicar el final de la instrucción, de igual manera para las secciones de declaración de variables y constantes, estas deben terminar con punto y coma (;) de lo contrario se marcará un error.

3. ESPECIFICACIÓN PARA DECLARAR VARIABLES

Ya hemos comentado sobre la sección de declaración de variables y algunas de sus reglas principales, cabe resaltar que es obligatorio incluir dicha sección en el programa. La forma de utilizar esta sección es la siguiente:

```
####Declaración de variables####  
var  
    entero : variable1;
```

Ilustración 2 Sección de declaración de variables

En la figura vemos un ejemplo en el cual declaramos una variable entera de nombre variable1, también podemos agregar valor al mismo tiempo que declaramos la variable, y sería de la siguiente forma:

```
####Declaración de variables####  
var  
    entero : variable1 := 12;
```

Ilustración 3 Declaración de variable1 con valor de 12

con esta instrucción le decimos al compilador que reserve espacio para un entero de nombre variable1 y le asigne el valor de 12, dicho valor podrá ser cambiado dentro del cuerpo principal del programa.

Resumiendo, la forma general para declarar variables es la siguiente:

```
####Declaración de variables####  
var  
    tipo : NombreVariable [:= Valor];
```

Ilustración 4 Forma general para declarar variables

Palabra clave	Descripción
tipo	Indica el tipo de variable a declarar.
:	Indica que se está tipificando una variable.
NombreVariable	Nombre que le damos a la variable
:=	Operador de asignación (se abordará en secciones adelante).
Valor	Valor que se le asigna a la variable.

La asignación de valor no es obligatoria por eso aparece entre corchetes, como se miró en el primer ejemplo (*ilustración 2*).

La expresión regular que usaremos para declarar variables es la siguiente:

Ejemplo de variable sin valor:

palRes = Palabras reservadas
simEsp = Símbolos especiales

letra = [a-zA-Z]
numero = [0-9]

variable = **(palRes:) letra (letra | numero)* simEsp**
variable = (palRes:) [a-zA-Z] ([a-zA-Z] | [0-9]) * ;

Ejemplo variable con valor:

palRes = Palabras reservadas
simEsp = Símbolos especiales

letra = [a-zA-Z]
numero = [0-9]

variable = **(palRes:)letra(letra|numero)*(:=([a-zA-Z][0-9])*)?simEsp**
variable = (palRes:)[a-zA-Z]([a-zA-Z][0-9])*(:=([a-zA-Z][0-9])*)?;

4. ESPECIFICACIÓN PARA DECLARAR CONSTANTES

Ya hemos comentado sobre la sección de declaración de constantes y algunas de sus reglas principales, cabe resaltar que es obligatorio incluir dicha sección en el programa. La forma de utilizar esta sección es la siguiente:

```
####Declaración de constantes####  
const  
    entero:const1 := 10;
```

Ilustración 5 Sección de declaración de constantes

La estructura de la instrucción puede resultar idéntica a declara una variable, y de echo lo es. No se trata de ningún error ni ningún tipo de problemas de diseño. Las estructuras son similares por el echo de que la declaración de variables y constantes solo se puede hacer en un lugar específico del programa, es decir, para declarar una constante basta con escribir la instrucción correcta en la sección de declaración de constantes. Se debe tener cuidado con esto dado que el compilador no ve ninguna diferencia entre las instrucciones, puede que deseemos declarar una constante, y nos equivoquemos de sección. Pero aparte de eso es correcto el uso de esta instrucción para ambas secciones.

Se puede rescatar una única diferencia entre una instrucción para declarar variables y constantes; para declara una constante es obligatorio asignarle un valor, dicho valor no podrá se alterado en el resto de la ejecución del programa.

Resumiendo, la forma general para declarar variables es la siguiente:

```
####Declaración de constantes####  
const  
    tipo : NombreConstante := Valor;
```

Ilustración 6 Forma general para declarar constantes

Palabra clave	Descripción
tipo	Indica el tipo de variable a declarar.
:	Indica que se está tipificando una variable.
NombreVariable	Nombre que le damos a la variable
:=	Operador de asignación (se abordará en secciones adelante).
Valor	Valor que se le asigna a la variable.

La expresión regular que usaremos para declarar constantes es la siguiente:

Ejemplo de constante:

palRes = Palabras reservadas
simEsp = Símbolos especiales

letra = [a-zA-Z]
numero = [0-9]

variable

=

(palRes:)letra(letra|numero)*(:=(letra|numero+(\.numero+?)?)*simEsp
variable = (palRes:) [a-zA-Z]([a-zA-Z][0-9])(:=([a-zA-Z][0-9]+(\.[0-9]+?)?)*);*

5. ESTRUCTURA CONDICIONAL SIMPLE (IF...ELSE)

La estructura condicional SI...SINO es una de las estructuras más útiles ya que nos permiten evaluar una expresión booleana y dependiendo el resultado se ejecuta un bloque de instrucciones para el caso positivo y otro para el caso negativo. Para esta estructura contamos con un grupo de palabras reservadas (si, entonces, iniciosi, finsi, sino, iniciosino, finsino) que nos indican el inicio y el final del bloque de instrucciones, así como la ubicación de la expresión a evaluar.

```
####Declaración de variables####
var
####Declaración de constantes####
const
####Cuerpo del programa####
inicio
    ####Estructura if####
    si 1 > 0 entonces
        iniciosi
            escribir('1 es mayor a 0');
        finsi
    sino
        iniciosino
            escribir('Esto no es posible...');
        finsino
fin
```

Ilustración 7 Ejemplo de estructura condicional IF...ELSE

Generalizando la estructura del ejemplo obtenemos la forma base de un IF...ELSE

```
####Estructura if####
si condición entonces
    iniciosi
        #Instrucciones para el caso verdadero
    finsi
    sino
        iniciosino
            #Instrucciones para el caso falso
        finsino
```

Ilustración 8 Forma general de la estructura condicional IF..ELSE

Palabra clave	Descripción
Si	Indica el inicio de la estructura condicional IF.
Condición	En este espacio se debe poner la expresión booleana a evaluar.
entonces	Indica al compilador donde termina la expresión booleana a evaluar.
iniciosi	Inicio del bloque de instrucciones para el caso positivo.
finsi	Fin del bloque de instrucciones del caso positivo.
sino	Indica al compilador que existe un bloque de instrucciones para el caso negativo.
iniciosino	Inicio del bloque de instrucciones para el caso negativo.
finsino	Fin del bloque de instrucciones del caso negativo.

El bloque *'else'* se puede omitir por eso se encuentra entre corchetes, pero si aparece la palabra reservada *'else'* debe aparecer las palabras reservadas *'beginelse'* y *'endelse'*, en ese orden o de lo contrario el compilador marcará error. Es igual para la palabra reservada *'if'* debe de ir precedida por las palabras reservadas *'beginif'* y *'endif'*, en ese orden si no se producirá un error. Esto se debe a que estas palabras reservadas (*'beginif'*, *'endif'*, *'beginelse'* y *'endelse'*) indican el inicio y el final de los bloques de instrucciones correspondientes a cada situación. El caso de la palabra reservada *'then'*, esta indica al compilador el final de la expresión a evaluar, si se omite se producirá un error.

6. ESTRUCTURA DE CONTROL (CICLO FOR)

La estructura FOR es un ciclo que se repite un numero determinado de veces; esta cantidad de repeticiones dependo de los valores especificados en su estructura. De esta manera podemos haces una cantidad 'n' de iteraciones. Dentro de esta estructura se pueden realizar 'n' cantidad de instrucciones. En la ilustración 9 se puede observar un ejemplo de uso del ciclo for.

```
####Declaración de variables####  
var  
####Declaración de constantes####  
const  
####Cuerpo del programa####  
begin  
    ####Estructura for####  
    para i := 1 hasta 10 hacer  
        iniciopara  
            escribir('Ciclo for');  
        finpara  
end
```

Ilustración 9 Ejemplo de ciclo FOR

Generaizando la estructura del ejemplo obtenemos la forma base del FOR:

```
####Estructura for####  
para i := número hasta numeroFinal hacer  
    iniciopara  
        escribir('Ciclo for');  
    finpara
```

Palabra clave	Descripción
para	Indica el inicio de la estructura condicional FOR
i := numero	Variable de control para las iteraciones del ciclo. Se le asigna el valor donde inicia la iteración.
hasta	Exclusiva del ciclo FOR se usa para definir el número final de iteraciones.
numeroFinal	Número final al que se requiere llegar la variable de control
hacer	Indica el final de la estructura
Iniciopara	Indica el inicio del bloque de instrucciones
finpara	Indica el final del bloque de instrucciones

En la especificación de la forma general es obligatorio el uso de todos los elementos descritos, agregando que la única forma de que funciones la estructura es indicando los valores con números enteros.

7. ESPECIFICACIÓN PARA IMPRIMIR EN PANTALLA Y LEER VARIABLES DESDE EL TECLADO

a. IMPRIMIR EN PANTALLA

Para imprimir en pantalla se hace uso de la función ‘*escribir*’. La forma correcta de utilizar esta función es la siguiente:

```
####Escribir en pantalla####  
    escribir ( cadena );
```

Ilustración 10 Imprimir en pantalla

Se llama a la función escribiendo su nombre seguido de la cadena a imprimir. El uso de paréntesis es obligatorio.

b. LEER VARIABLES DESDE TECLADO

Para leer variables desde teclado se hace uso de la función ‘*leer*’. La forma correcta de utilizar la función es la siguiente:

```
####Leer variable desde teclado####  
    leer ( variable );
```

Ilustración 11 Leer variable desde teclado

Se llama a la función escribiendo su nombre seguido de la cadena a imprimir. El uso de paréntesis es obligatorio. El valor que se le ingrese a la variable depende del tipo que se le asigne en la sección de declaración de variables, por lo tanto, no se le puede asignar una cadena a un entero.

8. PALABRAS RESERVADAS

Palabras reservadas	Estructura lógica a la que pertenece	Descripción
inicio		indica el inicio del cuerpo principal del programa.
fin		indica el fin del cuerpo principal del programa
var		indica el inicio de la sección de declaración de variables.
const		indica el inicio de la sección de declaración de constantes.
si	Ciclo si	indica el inicio del bloque condicional
entonces	Ciclo si	Aparece después de la expresión booleana indicando el final de esta.
iniciosi	Ciclo si	Inicio del bloque de instrucciones para el caso verdadero
finsi	Ciclo si	fin del bloque de instrucciones para el caso verdadero
sino	Ciclo si	Indica la existencia de un bloque de instrucciones para el caso falso.
iniciosino	Ciclo si	Inicio del bloque de instrucciones para el caso falso
finsino	Ciclo si	fin del bloque de instrucciones para el caso falso
para	Ciclo para	Indica el inicio de la estructura Para
hasta	Ciclo para	define el límite al que debe llegar la variable de control
hacer	Ciclo para	indica el final de la declaración de la estructura para
iniciopara	Ciclo para	Inicio del bloque de instrucciones del ciclo para
finpara	Ciclo para	Fin del bloque de instrucciones del ciclo para
escribir		Función para escribir en pantalla
leer		Función para leer valores mediante teclado
entero		tipo de valor nativo entero
real		tipo de valor nativo real
booleano		tipo de valor nativo booleano
cadena		tipo de valor nativo cadena

Para identificar las palabras reservados en nuestro programa usaremos la expresión regular que a continuación se muestra:

Se define:

$\Sigma = \{\text{inicio, fin, var, const, si, entonces, iniciosi, finsi, sino, iniciosino, finsino, para, hasta, hacer, iniciopara, finpara, escribir, leer, entero, real, booleano, cadena}\}$

PalabraReservada = .

9. OPERADORES ESPECIALES

Listado de operadores especiales que soporta el lenguaje de programación:

Operadores especiales	Descripción
+	suma aritmética
-	resta aritmética
*	multiplicación aritmética
/	división aritmética
>	mayor que, operador booleano
<	menor que, operador booleano
>=	mayor igual que, operador booleano
<=	menor igual que, operador booleano
==	igual que, operador booleano
!=	distinto que, operador booleano
:=	Operador de asignación
&	and, operador booleano
	or, operador booleano

10. SÍMBOLOS ESPECIALES

Listado de operadores especiales que soporta el lenguaje de programación:

Símbolo Especial	Descripción
;	Identifica el final de una instrucción
:	Símbolo especial para tipificar variables
“”	Las comillas dobles indican el inicio y final de una cadena de texto
()	Paréntesis, agrupador logico


```
####Declaración de variables####
var
    entero : variable1;

####Declaración de constantes####
const
    entero : const1 := 10;

####Cuerpo del programa####
inicio
    ####Estructura if####
    si condición entonces
        inicioif

        finif
    sino
        iniciosino

    finsino

    ####Estructura for####
    para valorInicial := numero hasta otroNumero hacer
        iniciopara

    finpara

    ####Entrada y salida desde teclado a pantalla####
    escribir ( cadena );
    leer( variable );

    ####Comentarios####
    #comentario de una línea
fin
```