



## Trabajo Práctico N° 9 Alcance, Entornos, Pasaje de Parámetros

### Observaciones:

En esta materia siempre hemos recomendado utilizar **nombres significativos** para los identificadores. Además **no permitimos** que en un procedimiento se utilicen **variables globales o de entornos no locales**.

Para ilustrar los conceptos relacionados a este práctico se utilizarán algunas de estas prácticas poco recomendables. *Sin embargo, es interesante observar como la utilización de estas prácticas "oscurece" la programación y dificulta el mantenimiento de los programas.*

**Ejercicio 1:** Defina las nociones de entorno de referencia de un bloque, alcance y visibilidad de un identificador.

**Ejercicio 2:** Dado el siguiente programa,

```
program Project1;
var numero: integer;
suma, producto : integer;

Procedure CalculaSumaProd(num: integer; var sumdig, prodig:integer);
begin
  sumdig := 0;
  prodig := 1;
  while num <> 0 do
    begin
      sumdig := sumdig + num mod 10;
      prodig := prodig * num mod 10;
      num := num div 10
    end;
end;

begin
  suma:=0; producto:=0;
  writeln('Ingrese un numero entero '); readln(numero);
  CalculaSumaProd(numero, suma, producto);
  writeln('La suma de digitos es ', suma);
  writeln('El producto de digitos es ', producto);
end.
```

- a) Realice la traza para número = 28976
- b) Reemplace el encabezado del procedimiento por  
`Procedure CalculaSumaProd(num: integer; sumdig, prodig:integer);`  
realice la traza nuevamente y compare los resultados con los obtenidos en el inciso anterior.



**Ejercicio 3:** Considere el siguiente programa en Lenguaje Pascal:

```
PROGRAM mayus;
CONST offset = ord('0');
VAR car: char;

FUNCTION EsMayuscula(c:char):boolean;
BEGIN {EsMayuscula}
  EsMayuscula := (c >= 'A') and (c <= 'Z')
END; {EsMayuscula}

FUNCTION EsMinuscula(car:char):boolean;
BEGIN {EsMinuscula}
  EsMinuscula := (car >= 'a') and (car <= 'z')
END; {EsMinuscula}

FUNCTION AMayuscula(c:char):char;
CONST offset = ord('A') - ord('a');
VAR car: char;
BEGIN {AMayuscula}
  IF esMayuscula(c) THEN
    car := c
  ELSE
    car := chr(ord(c) + offset);
  AMayuscula := car;
END; {AMayuscula}

FUNCTION EsVocal(c:char):boolean;
VAR minus:char;

  FUNCTION AMinuscula(c:char):char;
  CONST offset = ord('a') - ord('A');
  BEGIN {AMinuscula}
    IF esMayuscula(c) THEN
      minus := chr(ord(c) + offset)
    ELSE
      minus := c;
    AMinuscula := minus;
  END; {AMinuscula}

BEGIN {EsVocal}
  minus := AMinuscula(c);
  EsVocal := (minus = 'a') OR (minus = 'e') OR (minus = 'i') OR (minus = 'o') OR
(minus = 'u')
END; {EsVocal}

BEGIN {Ppal}
  writeln('Ingrese una frase terminada en punto "."');
  REPEAT
    read(car);
    IF EsMayuscula(car) OR EsMinuscula(car) THEN
      IF EsVocal(car) THEN write(AMayuscula(car))
      ELSE write(car);
    UNTIL (car = '.');
END. {Ppal}
```

a) Identifique los bloques del programa y los identificadores declarados en cada bloque.



b) Complete la siguiente tabla indicando para cada bloque su entorno de referencia (**no** incluya aquellos identificadores que quedan *ocultos*).

	mayus	EsMayuscula	EsMinuscula	AMayuscula	EsVocal	AMinuscula
<b>Entorno Global</b>						
<b>Entorno No Local</b>						
<b>Entorno Local</b>						

c) Como se afectaría la tabla del inciso anterior si se efectuaran sobre el programa anterior los siguientes cambios (considere los cambios en forma exclusiva, es decir, cuando considere algún inciso no efectúe ningún otro cambio de los otros incisos)

- a. Si se agregara una variable `AMinuscula` de tipo `char` al bloque `mayus` (bloque del programa `ppal`).
- b. Si se cambiara la función `AMayuscula` por la siguiente (observese que se deberá modificar la llamada):

```

FUNCTION AMayuscula:char;
CONST offset = ord('A') - ord('a');
BEGIN {AMayuscula}
  IF esMayuscula(car) THEN
    AMayuscula := car
  ELSE
    AMayuscula := chr(ord(car) + offset)
END; {AMayuscula}

```

- c. Si se cambiara la función `AMinuscula` por la siguiente:

```

FUNCTION AMinuscula(c:char):char;
FUNCTION offset(car:char):integer;
BEGIN
  offset := ord(car) + ord('a') - ord('A');
END;
BEGIN {AMinuscula}
  IF esMayuscula(c) THEN
    minus := chr(offset(c))
  ELSE
    minus := c;
  AMinuscula := minus;
END; {AMinuscula}

```



**Ejercicio 3:** Para cada uno de los siguientes casos mencione el alcance del identificador de constante PI.

<pre>PROGRAM caso1; <b>CONST PI = 3.14;</b>  PROCEDURE A; BEGIN END;  PROCEDURE B;   PROCEDURE C;   BEGIN   END;   PROCEDURE D;   BEGIN   END; BEGIN END;  BEGIN END.</pre>	<pre>PROGRAM caso2; <b>CONST PI = 3.14;</b>  PROCEDURE A; BEGIN END;  PROCEDURE B;   PROCEDURE C(<b>PI</b>:char);   BEGIN   END;   PROCEDURE D;   BEGIN   END; BEGIN END;  BEGIN END.</pre>	<pre>PROGRAM caso3; <b>CONST PI = 3.14;</b>  PROCEDURE A; BEGIN END;  FUNCTION <b>PI</b>(X:char):boolean;   PROCEDURE C;   BEGIN   END;   PROCEDURE D(<b>PI</b>:char);   BEGIN   END; BEGIN   PI := ... END; BEGIN END.</pre>
---	---	---

**Ejercicio 4:** Dado el siguiente programa en Pascal

```
PROGRAM Entorno;
CONST Z=100;
VAR A, B:INTEGER;
    EsPar:BOOLEAN;

PROCEDURE Multiplica(VAR X:INTEGER; Y:INTEGER);forward;

  PROCEDURE Cubo(VAR A:INTEGER);
  BEGIN
    A:= A * A * A
  END;

  PROCEDURE Multiplica(VAR X:INTEGER; Y:INTEGER);
  VAR Z:INTEGER;
  FUNCTION EsPar(A:INTEGER):BOOLEAN;forward;

    FUNCTION EsMayor(A,B:INTEGER):BOOLEAN;
    BEGIN
      EsMayor:= A > B
    END;

    FUNCTION EsPar(A:INTEGER):BOOLEAN;
    BEGIN
      EsPar := (A mod 2 = 0)
    END;

  BEGIN
    X:= X + 10;
    IF EsMayor(X,Y) THEN Cubo(Y)
  END;

BEGIN
  A:= 2; B:= 6;
  Multiplica(A, B);
  WRITELN(A, ' - ', B);
  Cubo(B);
  WRITELN(A, ' - ', B);
  readln;

END.
```



a) Realice la traza del programa ¿Qué se muestra por pantalla como resultado de la ejecución del programa Entorno?

- |                    |                     |                       |                            |                              |
|--------------------|---------------------|-----------------------|----------------------------|------------------------------|
| 1) 2 - 6<br>2 - 6  | 3) 2 - 6<br>216 - 6 | 5) 12 - 6<br>12 - 216 | 7) 2 - 216<br>2 - 216      | 9) 2 - 216<br>10077696 - 216 |
| 2) 2 - 6<br>2 - 18 | 4) 12 - 6<br>12 - 6 | 6) 12 - 6<br>216 - 6  | 8) 2 - 216<br>2 - 10077696 | 10) 2 - 18<br>8 - 18         |

b) ¿Cuál es el alcance del identificador A definido en el ambiente Entorno?

- |  |                     |                           |  |
|--|---------------------|---------------------------|--|
| 1) Entorno,<br>Multiplica, Cubo,<br>EsPar, EsMayor | 2) Entorno,<br>Cubo | 3) Entorno,<br>Multiplica | 4) Entorno,<br>Multiplica,<br>EsPar, EsMayor |
|--|---------------------|---------------------------|--|

c) ¿En qué ambientes de referenciamiento es visible el identificador Z definido en el ambiente Multiplica?

- |  |         |                                  |               |
|--|---------|----------------------------------|---------------|
| 1) Entorno,<br>Multiplica, Cubo,<br>EsPar, EsMayor | 2) Cubo | 3) Multiplica,<br>EsPar, EsMayor | 4) Multiplica |
|--|---------|----------------------------------|---------------|

¿El identificador EsPar definido en el ambiente Entorno es visible en el ambiente Multiplica? Justifique su respuesta