

Algoritmos y Programación I - 2025

Práctica 2 – Modularización con parámetros

Objetivos:

- Reconocer la importancia de la modularización como estrategia en la resolución de problemas.
- Distinguir entre los dos tipos de módulos en Pascal (funciones y procedimientos). Identificar cuál es más conveniente en cada problema.
- Aplicar el mecanismo de comunicación de pasaje de parámetros en la resolución de problemas.

Aclaración: si bien en esta práctica se presentan soluciones utilizando variables globales, el objetivo de las mismas es comprender su funcionamiento y mostrar algunos de los problemas asociados con su uso. A la hora de resolver ejercicios en las siguientes prácticas, **NO** se deberán utilizar variables globales.

1. Dado los siguientes programas indicar qué imprime cada uno.

<p>a.</p> <pre> program alcance1; var a,b: integer; procedure uno; var b: integer; begin b := 12; writeln(b); end; begin a:= 4; b:= 5; uno; writeln(b, a); end.</pre>	<p>b.</p> <pre> program alcance2; var a,b: integer; procedure dos; begin b := 9; writeln(b); end; begin a:= 4; b:= 8; dos; writeln(b, a); end.</pre>	<p>c.</p> <pre> program alcance3; var a: integer; procedure tres; var b: integer; begin b:= 5; writeln(b); end; begin a:= 6; tres; writeln(a, b); end.</pre>
---	--	--

2. Dado el siguiente programa, indicar cuál es el error.

<pre> program alcance4; function cuatro: integer; begin cuatro:= 4; end;</pre>	<pre> var a: integer; begin cuatro; writeln(a); end.</pre>
--	--

3. Dado el siguiente programa:

```

program alcanceYFunciones;
var
  suma, cant: integer;
function calcularPromedio: real;
var
  prom: real;
begin
  if (cant = 0) then
    prom := -1
  else
    prom := suma / cant;
end;
begin {programa principal}
  readln(suma);
  readln(cant);
  if (calcularPromedio <> -1) then begin
    cant := 0;
    writeln('El promedio es: ', calcularPromedio)
  end
  else
    writeln('Dividir por cero no parece ser una buena idea');
end.

```

- a. La función calcularPromedio calcula y retorna el promedio entre las variables globales suma y cant, pero parece incompleta. ¿Qué debería agregarle para que funcione correctamente?
- b. En el programa principal, la función calcularPromedio es invocada dos veces, pero esto podría mejorarse. ¿cómo debería modificarse el programa principal para invocar a dicha función una única vez?
- c. Si se leen por teclado los valores 48 (variable suma) y 6 (variable cant), ¿qué resultado imprime el programa?
Considere las tres posibilidades:

- i) El programa original
- ii) El programa luego de realizar la modificación del inciso a)
- iii) El programa luego de realizar las modificaciones de los incisos a) y b).

4. a. Realice un módulo que reciba dos números enteros y retorne la suma de los mismos.
b. Implemente un programa que invoque al módulo del inciso a, e informe lo pedido.
5. a. Realice un módulo que lea de teclado números enteros hasta que llegue un valor negativo y retorne cuál fue el número **impar** más alto.
b. Implemente un programa que invoque al módulo del inciso a, e informe lo pedido.
6. a. Escriba un procedimiento que lea la altura de una cantidad de personas y devuelva el promedio de éstas. La cantidad de personas se recibe como parámetro.
b. Escriba un programa que procese la altura de 48 personas utilizando el módulo desarrollado en a) e informe el resultado.
7. Dada la siguiente función marque las invocaciones a dicha función que **considere válidas**:

```

function cuadrado(x:integer): integer;

begin
  cuadrado:= x*x;

end

```

- a. Write(cuadrado(8));
 - b. c:= cuadrado(8); Write(cuadrado);
 - c. If (cuadrado = 64) then
 - d. Write('8*8=64');
 - e. cuadrado(8);
 - f. c:= cuadrado(8); Write (c);
 - g. cuadrado(8, c); Write (c);
 - h. If (cuadrado(8) = 64) then
8. Dado el siguiente programa:

```
program anidamientos;
  procedure leer;
  var
    letra: char;

    function analizarLetra: boolean
    begin
      if (letra >= 'a') and (letra <= 'z') then
        analizarLetra := true;
      else
        if (letra >= 'A') and (letra <= 'Z') then
          analizarLetra := false;
        else
          analizarLetra := false;
        end;
      end;
    end; {fin de la funcion analizarLetra}
  begin
    readln(letra);
    if (analizarLetra) then
      writeln('Se trata de una minúscula');
    else
      writeln('Se trata de una mayúscula');
    end;
  end; {fin del procedure leer}
var
  ok: boolean;
begin {programa principal}
  leer;
  ok := analizarLetra;
  if ok then
    writeln('Gracias, vuelva prontoss');
  end.
```

- a. La función **analizarLetra** fue declarada como un submódulo dentro del procedimiento leer. Pero esto puede traer problemas en el código del programa principal.
 - i) ¿Qué clase de problema encuentra?
 - ii) ¿cómo se puede resolver el problema para que el programa compile y funcione correctamente?
 - b. La función **analizarLetra** parece incompleta, ya que no cubre algunos valores posibles de la variable letra.
 - i) ¿De qué valores se trata?
 - ii) ¿Qué sucede en nuestro programa si se ingresa uno de estos valores?
 - iii) ¿Cómo se puede resolver este problema?
9. Responda las preguntas en relación al siguiente programa:

```
program Ejercicio3;
  procedure suma(num1: integer; var num2:integer);
  begin
    num2 := num1 + num2;
    num1 := 0;
  end;
var
  i, x: integer;
begin
  read(x); { Leo la variable x }
  for i:= 1 to 5 do
    suma(i,x);
  write(x); { imprimo las variable x }
end.
```

- a. ¿Qué imprime si se lee el valor 10 en la variable x ?
 - b. ¿Qué imprime si se lee el valor 10 en la variable x y se cambia el encabezado del procedure por:
procedure suma(num1: integer; num2:integer);
 - c. ¿Qué sucede si se cambia el encabezado del procedure por:
procedure suma(**var** num1: integer; **var** num2:integer);
10. Realizar un programa modularizado que lea datos de 100 productos de una tienda de ropa. Para cada producto debe leer el precio y código. Informar:
- Código de los dos productos más baratos.
 - Precio promedio.
11. El factorial de un número n se expresa como **n!** y se define como el producto de todos los números desde 1 hasta n. Por ejemplo, el factorial de 6 o 6! equivale a 6*5! es igual a 1*2*3*4*5*6 que equivale a 720. Escriba una función que reciba un número n y retorne su factorial.
12. Escriba un módulo que reciba 2 números enteros i y n, y calcule la potencia enésima de i (i^n).