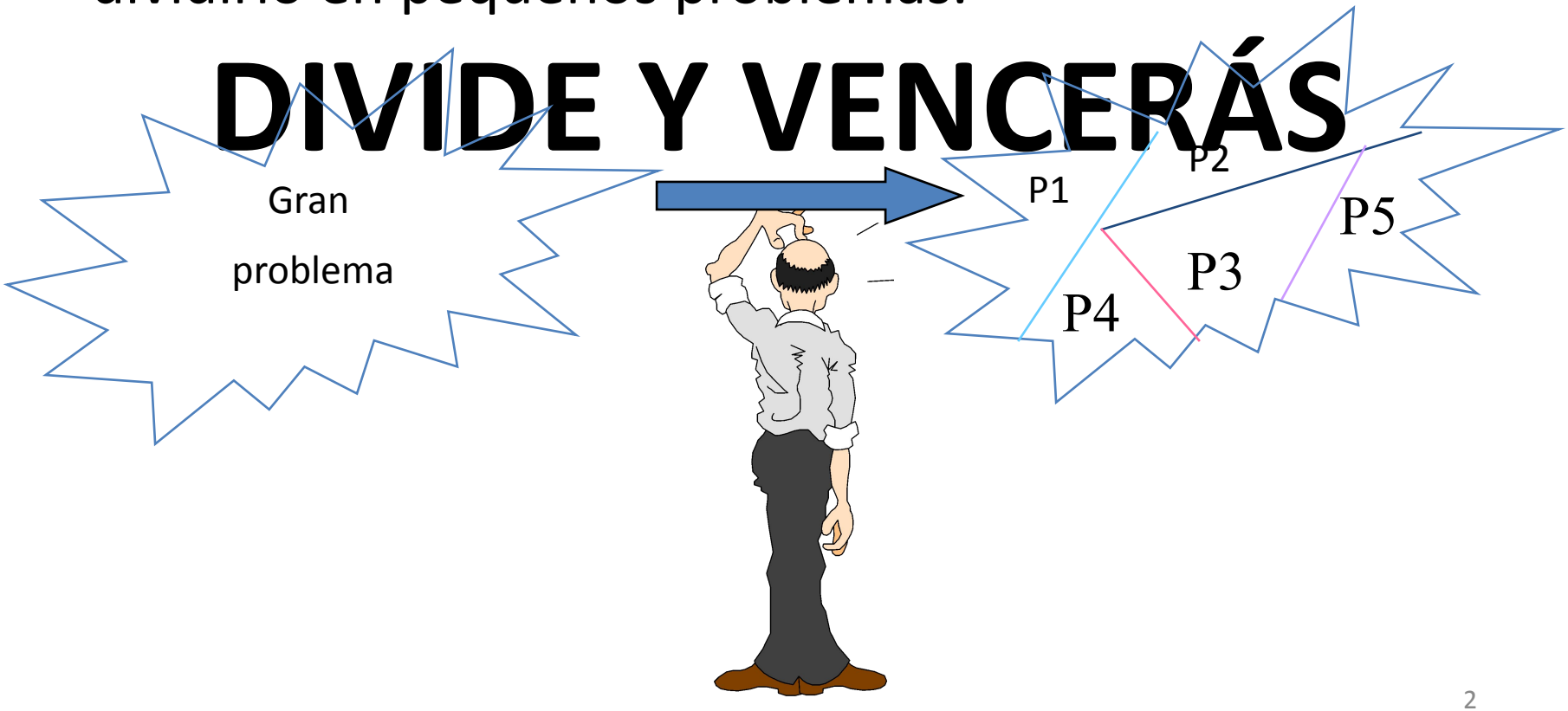
Procedimientos y Funciones

Programación I

Cuando nos encontramos frente a un gran problema, lo mejor que podemos hacer para solucionarlo es dividirlo en pequeños problemas.

DIVIDE Y VENCERÁS



Procedimientos y Funciones

Disponemos de dos herramientas básicas para realizar programación descendente:

- los procedimientos (procedure)
- las funciones (function),

a estos módulos también son referidos genéricamente con el término de subprogramas.

Procedimientos y Funciones

Procedimientos:

- Deben realizar una tarea específica.
- Pueden recibir cero o más valores del programa que llama y devolver cero o más valores a dicho programa llamador.
- Se los invoca como una sentencia.

Funciones:

- También realizan una tarea específica
- Devuelven un valor a la unidad de programa que los referencia.
- Se los invoca utilizando alguna sentencia que nos permita recibir el valor devuelto.

Declaración de un procedimiento (1):

```
procedure nombreProc;

//Declaracion local de variables y subprogramas

begin

  //cuerpo del procedimiento

end;
```

Declaración de un procedimiento (2):

```
procedure nombreProc (lista de parametros formales);
//Declaracion local de variables y subprogramas
begin
  //cuerpo del procedimiento
end;
```

Procedimientos Ejemplo

```
program Ejemplos;
{$APPTYPE CONSOLE}

var
  x, y: integer;

  procedure resta;
  begin
    x := x - 1;
    y := y - 1;
  end;

begin

  x:= 5;
  y:= 10;
  resta;
end.
```

Procedimientos Ejemplo

```
program Ejemplos;
{$APPTYPE CONSOLE}
var
  x, y, z, w: integer;

  procedure resta(a: integer, var b: integer);
  begin
    a := a - 1;
    b := b - 1;
  end;

begin

  x:= 5;      y:= 10;       z := 99;        w := 23;
  resta(x, y);
  resta(z, w);
end.
```

Lista de parámetros formales

Lista de parámetros actuales

Parámetros Formales

- **Parámetros por referencia**
- Cuando se pasa una variable a un procedimiento como parámetro por referencia, los cambios que se efectúan sobre dicha variable dentro del procedimiento se mantienen, incluso después de que haya finalizado el procedimiento. Los cambios producidos en parámetros por referencia son permanentes, ya que no se pasa al procedimiento el valor de hay dentro de la variable, sino la dirección de memoria de la variable.
- **Parámetros por valor**
- Los parámetros por valor son diferentes a los parámetros por referencia, cuando se pasa un parámetro por valor a un procedimiento se guarda en memoria una copia temporal de la variable, Dentro del procedimiento solo se utiliza la copia, la original nunca se toca.

Procedimientos Ejemplo

```
program Ejemplos;
var
  x,y,z,w: integer;

  procedure resta(a: integer, var b: integer);
  begin
    a := a - 1;
    b := b - 1;
  end;

begin

  x:= 5;      y:= 10;       z:= 99;       w := 23;
  resta(x, y);
  resta(z, w);
end.
```

Declaración de una función:

```
function nombreFuncion(A: integer): tipo;
  // declaraciones locales y subprogramas
begin
  // cuerpo de la Función
  nombreFuncion := valor de la función
end;
```

Funciones
Ejemplo

```
program Ejemplos;
var
  x, y, z: real;

  function promedio(a, b: real): real;
  begin
    promedio := (a + b) / 2;
  end;

begin

  x:= 5,1;
  y:= 10,3;
  z := promedio(x, y);

end.
```

Ámbito de las Variables

```
program AmbitoVariables;
procedure ejemplo();
var a:integer;
begin
    a:= 100;
   writeln ('a=',a);
end;
var a:integer;
begin
  a:=1;
  writeln ('a=',a);
  ejemplo();
end.
```

Variables locales y globales

- Según visto podemos definir un procedimiento dentro de otro.
- Entonces:
  - Las variables globales del programa pueden ser usadas en cualquier procedimiento o función.
  - Las variables definidas en un módulo pueden ser usadas por cualquier módulo definido dentro de este módulo

Consideraciones Especiales

Si utilizamos variables globales en los módulos

- corremos el riesgo que en un módulo “A” se modifique una variable global que el módulo “B” necesita sin modificar. Por tanto la “convivencia” puede ser difícil.
- dificultamos la legibilidad del código
- perdemos reutilización del código

POR ESO SE RECOMIENDA NO HACER USO DE VARIABLES GLOBALES