COMPUTACIÓN I
TEMA 5.
Subprogramas o funciones. Alcance de variables. Pasaje de parámetros por valor

Prof. Mireya Morales

CONTENIDO

- Instrucciones break y continue
- Importancia del uso de subalgoritmos o subprogramas
- Definición de Funciones
- Definición de Procedimientos
- Alcance de las variables: Globales y Locales
- Comunicación de subprogramas a través del pasaje de parámetros.
- Paso de parámetros por valor y por referencia.

Instrucción break

En ocasiones es conveniente salir del bucle sin tomar en cuenta la condición. Ejemplo:

```
while((caracter=getchar()) != 'S' )
  {
    if (caracter = = '.')
        break;
    printf("has introducido %c",caracter);
  }
```

Instrucción break

Si esta instrucción no se usa adecuadamente, puede perturbar la programación estructurada, ella es recomendada cuando ocurre una excepción. Una mejora del programa anterior sería:

```
while((caracter=getchar()) != 'S' && caracter !='.')
  {
     printf("has introducido %c",caracter);
  }
```

Instrucción continue

En algunos casos el cuerpo de un bucle puede resultar complejo y es posible que eventualmente no se requiera ejecutar una parte de las instrucciones, aunque se desea permanecer dentro de él. Ejemplo:

```
while((caracter=getchar()) != 'S')
  {
    If ( caracter= = '.')
       continue;
    printf("has introducido %c",caracter);
  }
```

Instrucción continue

Una versión mejorada del código anterior es:

```
while((caracter=getchar()) != 'S')
  {
     if ( caracter! = '.')
        printf("has introducido %c",caracter);
   }
```

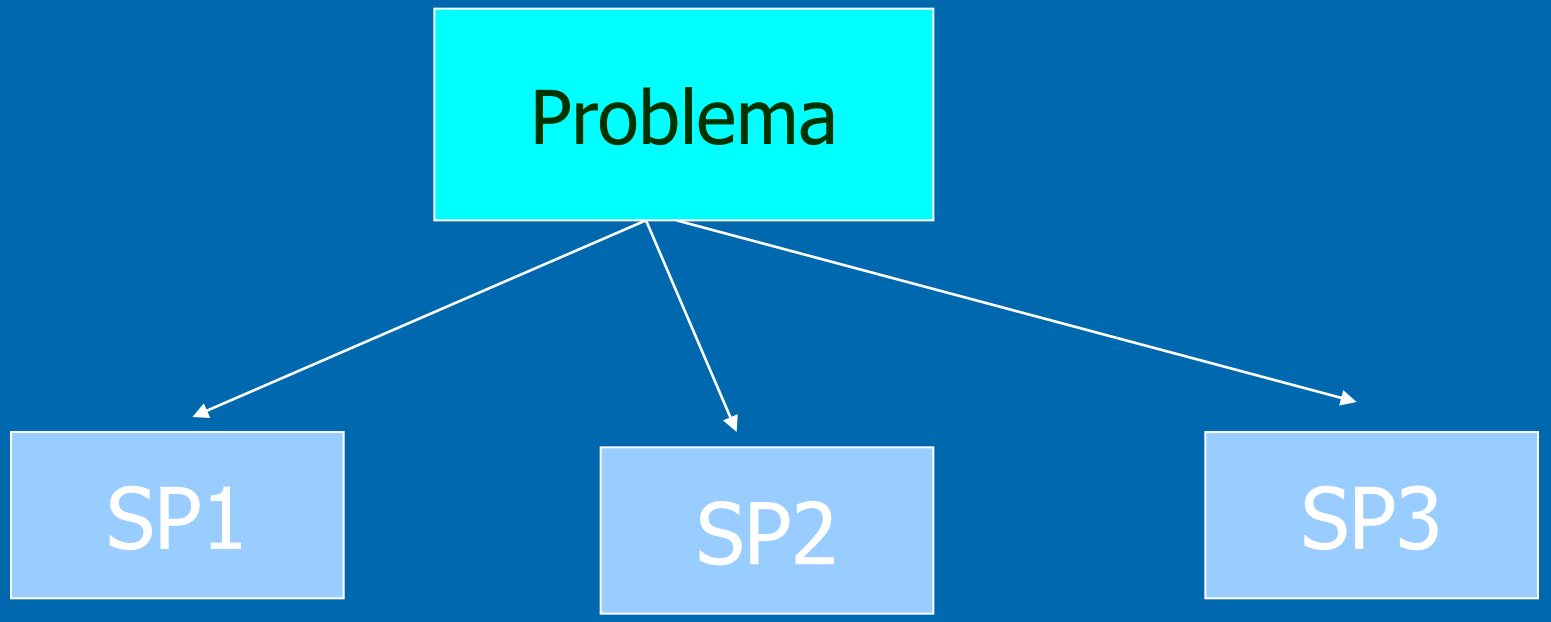
Importancia del uso de subalgoritmos o subprogramas

- Los subalgoritmos permiten a los programadores desarrollar soluciones de problemas complejos, a través del método descendente (top-down).
- Representan unidades de programas diseñados para ejecutar una tarea específica.
- Los subprogramas se escriben 1 sola vez, pero pueden ser referenciados en diferentes puntos de un programa

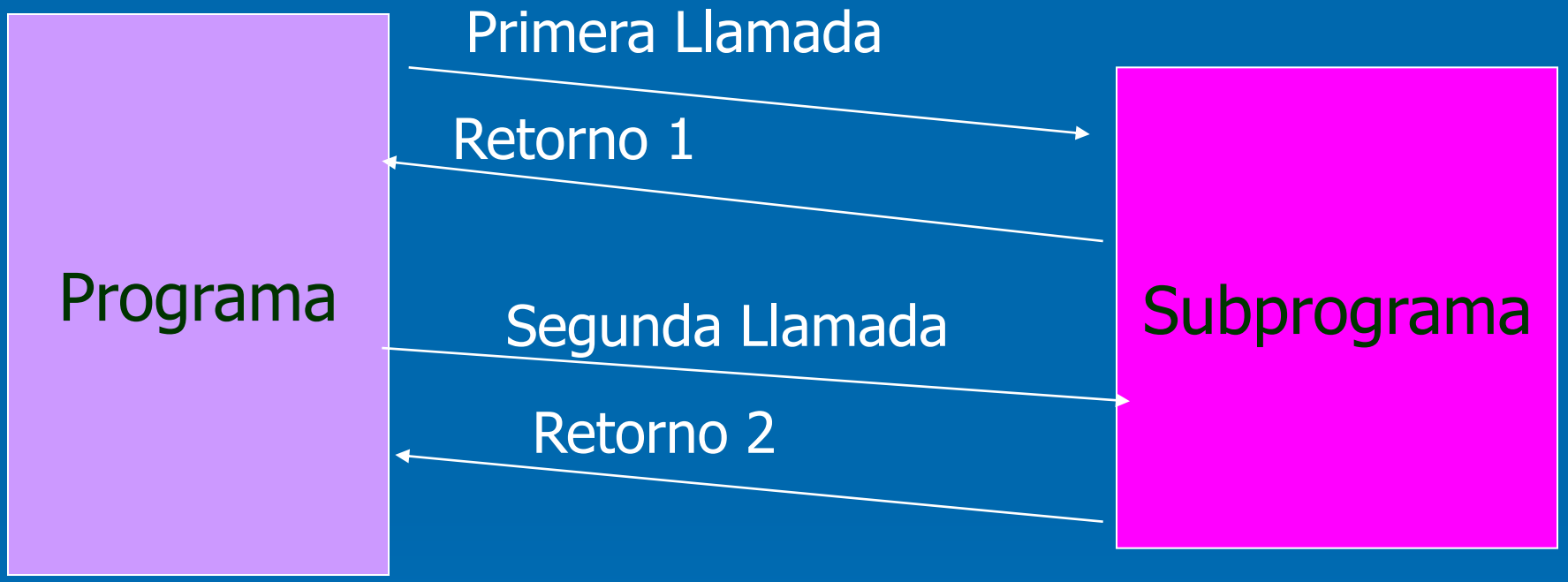
Importancia del uso de subalgoritmos o subprogramas

- Con el uso de los subprogramas se facilita la localización de un error.
- Los programas son más fáciles de comprender, debido a la independencia entre las unidades de programas.

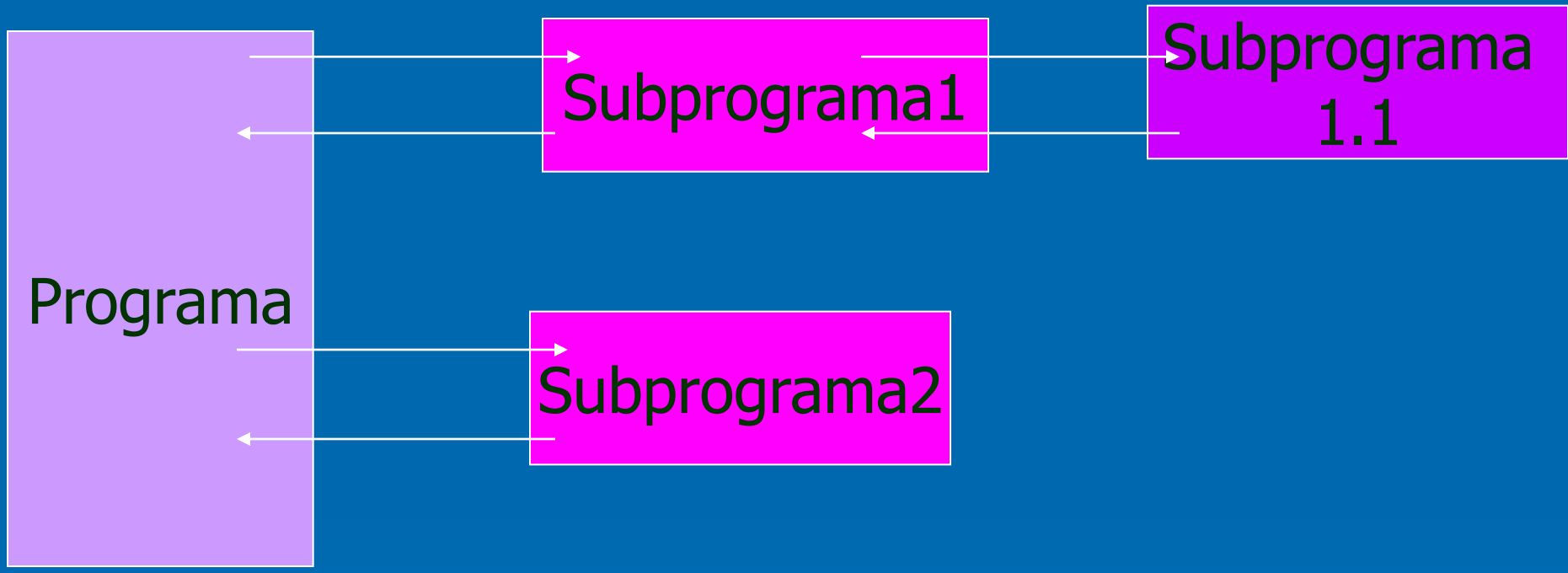
Importancia del uso de subalgoritmos o subprogramas



El programa principal invoca al subprograma



Diferentes niveles de subprogramas



Funciones

- Representan rutinas de programas que son invocadas desde algún otro programa.
- Son referenciadas a través de un nombre y una lista de parámetros **actuales** o **reales**.

Declaración de Funciones

<tipo de resultado> función <nom_fun> (lista de parámetros)

[declaraciones locales]

inicio

<acciones> /*cuerpo de la función*/

devolver (<expresion>)

fin_función

La llamada es mediante una instrucción con la siguiente estructura:

X=nombre_función(lista de parámetros):

Ejemplo Declaración de Funciones

f(x)=x/(1 +x*x)

```
<real> función F(x)
inicio
   devolver (x/(1+x*x))
fin_funcion
```

Ejemplo: Valor absoluto de un número

```
Algoritmo Calculo_valor_absoluto
   var X,Y entero
   inicio
      leer X
      Y=valor_absoluto(X)
      escribir El valor absoluto de X es Y
   fin
```

Ejemplo: Valor absoluto de un número

```
entero valor_aboluto(entero:N)
var Z entero
Inicio
   si N<0 entonces
      Z= N*(-1)
   sino
     Z=N
   fin_si
   devolver Z
fin_funcion
```

Procedimientos

- Es un subprograma que ejecuta un proceso específico.
- Ningún valor está asociado con el nombre del procedimiento.
- Cuando un procedimiento es invocado se ejecutan las acciones que lo definen y luego se devuelve el control a la instrucción siguiente a la llamada.
- A diferencia de las funciones no devuelven valores.

Procedimiento

Procedimiento nombre [(lista de parámetros formales)]

<acciones>

Fin_procedimiento

La llamada es mediante una instrucción con la siguiente estructura:

[llamar_a] nombre [(lista de parámetros)]

Procedimiento. Cálculo de la división de dos números para obtener cociente y resto

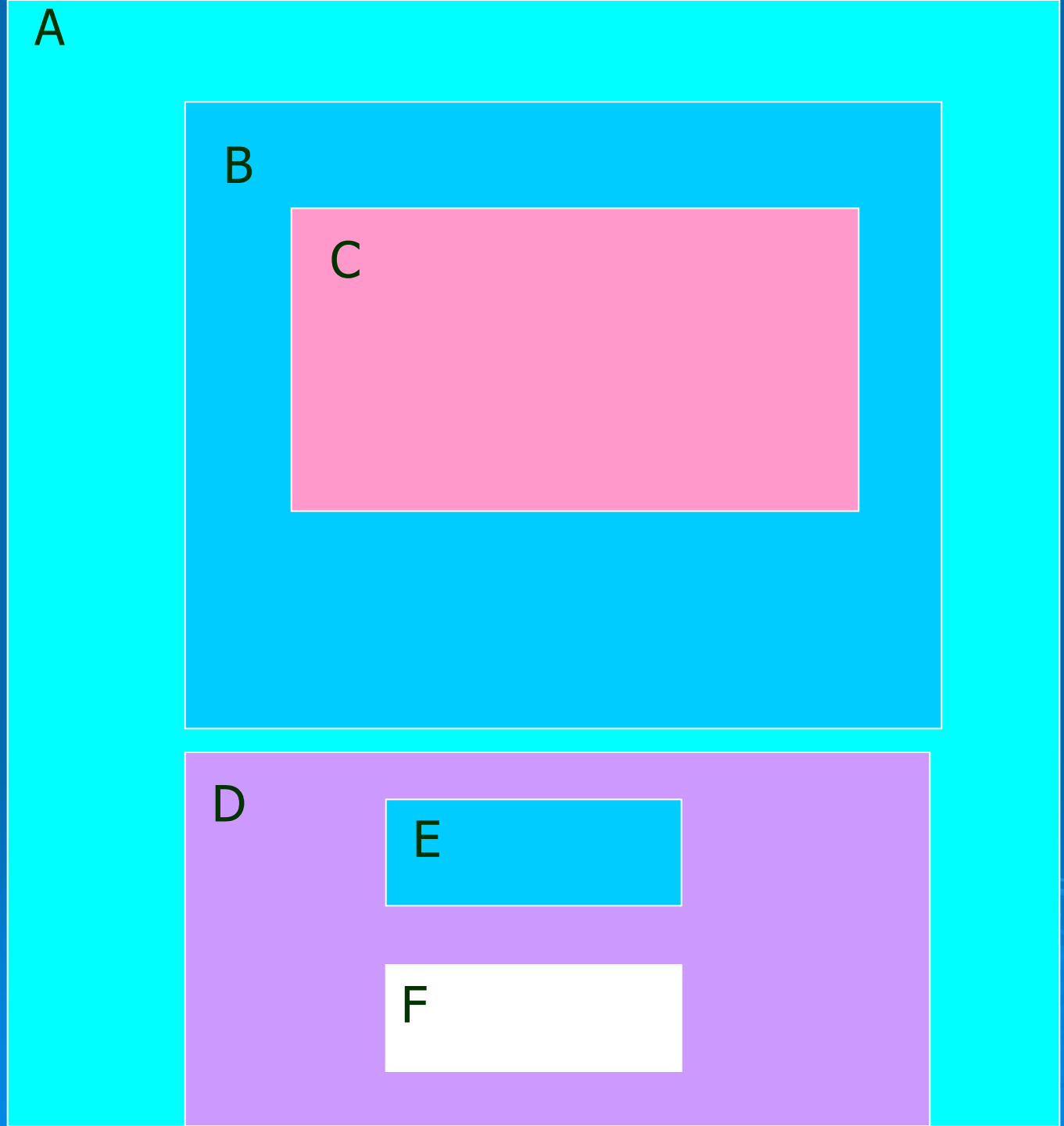
```
Algoritmo calculo_cociente_resto
Var M,N,P,Q entero
inicio
   leer M
   leer N
   llamar division (M,N,P,Q)
fin
```

Procedimiento. Cálculo de la división de dos números para obtener cociente y resto

```
Procedimiento division(entero:D,d,C,R)
inicio
    C = D div d
    R = D-C*d
    escribir C,R
fin_procedimiento
```

Alcance de Variables

- **Variables locales**: Es aquella declarada dentro de un subprograma y es distinta de las variables que tengan el mismo nombre en cualquier parte del programa principal. Cuando se hace referencia a una variable con el mismo nombre que otra dentro de un programa, se refiere a una posición diferente de memoria.
- **Variables Globales**: Es aquella que está declarada en el programa principal y se puede referenciar desde cualquier parte del programa, inclusive desde otros subprogramas.


A
B
C
D
E
F

Paso de Parámetros

- **Por valor**: Se pasa una copia del parámetro original al subprograma llamado. Los cambios a la copia no afectan el valor de la variable original en el lugar donde se hace la llamada.

- **Por referencia**: El invocador le da al subprograma llamado la capacidad de acceder directamente a la posición de memoria donde se encuentra el dato y modificarlo si se desea.