

Capítulo 7

Parámetros de Entrada/salida

Objetivos

Continuando con los mecanismos de comunicación entre módulos se incorporarán en este capítulo los parámetros de entrada/salida que, como su nombre lo indica, permiten realizar un intercambio de información entre módulos, en ambos sentidos.

Este tipo de parámetros, si bien puede utilizarse en reemplazo de los parámetros de entrada, es recomendable utilizarlos solo en aquellos casos en que la comunicación entre los módulos lo justifique. De esta manera se reducirá la aparición de errores no deseados.

Temas a tratar

- ✓ Introducción
- ✓ Ejemplos
- ✓ Otro uso de los Parámetros de Entrada/Salida.
- ✓ Conclusiones

7.1 Introducción

Un módulo utiliza un parámetro de entrada/salida cuando necesita recibir un dato, procesarlo y devolverlo modificado. También se utiliza para que un módulo pueda darle información al módulo que lo llamó.

El parámetro de entrada/salida permite realizar ambas operaciones sobre el mismo parámetro, ampliando de esta forma las posibilidades de comunicación.

Si bien este aspecto puede parecer ventajoso en primera instancia, es importante considerar que el uso de este tipo de parámetros resta independencia al módulo llamado ya que su funcionamiento depende de la información recibida.

7.2 Ejemplos

Ejemplo 7.1: Programe al robot para que informe la cantidad total de flores que hay en la avenida
4. No se debe modificar la cantidad de flores de cada esquina.

```
programa Cap7Ejemplo1
subprogramas
    procedimiento SumarFloresEsquina (es flores:numero ) (2)
    variables
        aux: numero
    comenzar
        aux:= 0
        mientras HayFlorEnLaEsquina
            comenzar
                tomarFlor
                aux:= aux + 1
                flores:= flores + 1
            fin
        repetir aux
            depositarFlor
    fin

variables
    totalFlores : numero
comenzar
    iniciar
    Pos (4,1)
    totalFlores := 0
    repetir 9
        comenzar
            SumarFloresEsquina (totalFlores)                (1)
            Mover
        fin
        SumarFloresEsquina (totalFlores)                (3)
    Informar (totalFlores)
fin
```

En Cap7Ejemplo1, el procedimiento SumarFloresEsquina recibe, en cada invocación, el total de flores encontradas hasta el momento y sobre este valor, continúa acumulando las flores. Esto puede verse en (1), donde al realizar la invocación se utiliza a totalFlores como parámetro. En (2) se especifica que este parámetro es de entrada/salida. Es decir, al comenzar la ejecución del procedimiento SumarFloresEsquina, este recibe en flores el valor del parámetro actual totalFlores, sobre este valor continúa acumulando la cantidad de flores encontradas y al finalizar, el valor del parámetro formal flores será devuelto al programa principal a través de totalFlores, reflejando de esta forma las modificaciones realizadas dentro del procedimiento.

Expresión de Problemas y Algoritmos

Carreras: Lic. en Sistemas - AUS – IDEI -UNTDF

Es importante ver que el objetivo del procedimiento SumarFloresEsquina es modificar la cantidad de flores encontradas hasta el momento sumándole la cantidad de flores de la esquina actual.

Ejemplo 7.2: Programe al robot para que recorra todas las avenidas de la ciudad e informe la cantidad total de flores encontradas.

La descomposición Top-Down del problema se muestra en la figura 7.1

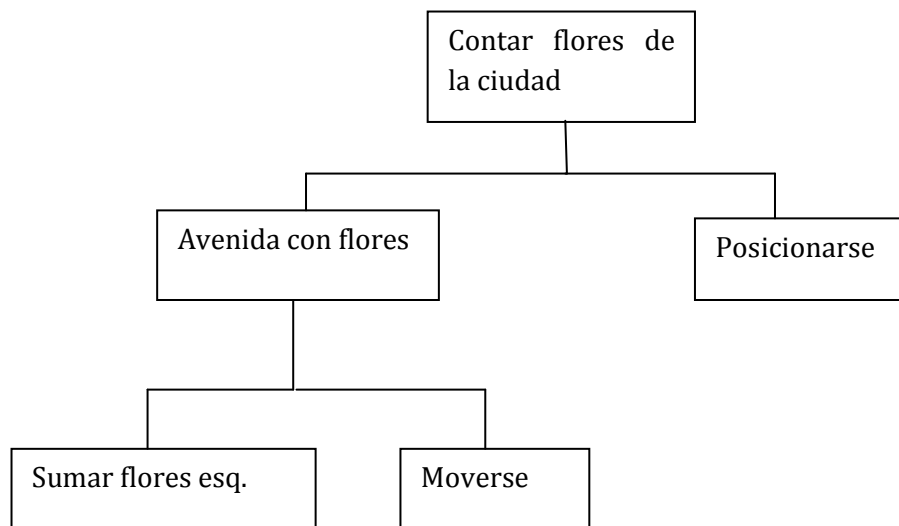


Fig. 7.1: Top-Down del ejemplo 7.2

El algoritmo es de la forma:

```
{Para cada avenida de la ciudad}
  {Recorrer la avenida incrementado la cant. de flores encontradas}
  {Posicionarse en la próxima avenida}
{Recorrer la ultima avenida}
{Informar el total de flores encontradas}
```

El programa utiliza dos módulos: uno para contar las flores de la esquina que ya fue definido en el ejemplo 7.1 y otro para recorrer la avenida.

El procedimiento que recorre la avenida es el siguiente:

```
procedimiento AvenidaConFlores( es Total : numero )
variables
  cuantas : numero
comenzar
  repetir 9
    comenzar
      SumarFloresEsquina (cuantas)
```

```
Total := Total + cuantas
Mover
fin
{ esq. de la calle 10}
SumarFloresEsquina (cuantas)
Total := Total + cuantas
fin
```

Como puede verse, posee un parámetro de entrada/salida para registrar el total de flores del recorrido. Cada vez que el procedimiento es invocado recibe como entrada la cantidad de flores encontradas hasta el momento, sobre este valor agrega las flores de esta avenida y lo devuelve modificado.

El programa completo es el siguiente:

programa Cap7Ejemplo2V1

subprogramas

procedimiento *SumarFloresEsquina* (es flores :numero)

comenzar

{ este procedimiento fue definido en el ejemplo 7.1 }

fin

procedimiento *AvenidaConFlores*(es Total : numero)

{ este es el procedimiento definido más arriba }

variables

TotFlores : numero

comenzar

iniciar

TotFlores := 0

(1)

repetir 9

comenzar

AvenidaConFlores(TotFlores)

(2)

Pos(PosAv + 1, 1)

fin

AvenidaConFlores(TotFlores)

Informar(TotFlores)

(3)

fin

En este código puede verse que, el determinar correctamente la cantidad de flores de la ciudad es responsabilidad tanto del programa principal como del procedimiento *AvenidaConFlores*.

En (1) el programa principal asigna el valor 0 a la variable *TotFlores* como forma de representar que hasta el momento no se ha encontrado ninguna flor. En (2), al producirse la primer invocación al procedimiento *avenida*, se le envía el valor 0 que es recibido por el parámetro formal de entrada/salida, *total*. Durante la ejecución del procedimiento *AvenidaConFlores*, *total* se va incrementando con las flores encontradas en esa avenida. Al finalizar la avenida, se asigna este

valor sobre el parámetro actual, *TotFlores*, permitiendo que el programa principal conozca la cantidad de flores encontradas en la avenida 1.

Luego de posicionarse en la avenida 2 se invoca nuevamente al procedimiento enviándole la cantidad de flores encontradas en la avenida 1. El procedimiento recibe esta cantidad y la

Expresión de Problemas y Algoritmos

Carreras: Lic. en Sistemas - AUS – IDEI -UNTDF

incrementa con el total de flores de la avenida 2. Al finalizar, asigna nuevamente en TotFlores este valor permitiendo que el programa principal conozca la cantidad de flores encontradas en las primeras dos avenidas.

Esto se repite para las 8 avenidas restantes por lo cual en (3) se informará la cantidad de flores encontradas en todas las avenidas de la ciudad.

Ejemplo 7.3: Modifique la implementación del ejemplo 6.4 para que el robot informe al finalizar su recorrido, la cantidad total de vértices que tienen flores (al menos una).

El programa que sigue muestra la solución implementada en DaVinci:

```
programa Totaldeflores
subprogramas
  procedimiento Rectangulo (en base :numero; en altura :numero; es
cantidad:numero)
comenzar
repetir 2
  comenzar
  si(HayFlorEnLaEsquina)
    cantidad := cantidad + 1
    {realiza un lado con tantas cuadradas como indica altura}
  repetir altura
    mover
  si (HayFlorEnLaEsquina)
    cantidad := cantidad + 1
  derecha
  {realiza un lado con tantas cuadradas como indica base}
  repetir base
    mover
  derecha
  fin
fin

variables
  ancho : numero
  alto : numero
  cantVertices : numero
comenzar
  iniciar
  {el primer rectángulo tiene ancho 9 y alto 5}
  ancho := 9
  alto := 4
  cantVertices := 0
  repetir 3
    comenzar
    Rectangulo (ancho, alto, cantVertices)
    {posición inicial del próximo rectángulo}
    pos(PosAv+1, PosCa + alto)
    ancho := ancho - 2
    alto := alto -1
    fin {muestra la cantidad de flores encontradas en todos los vértices de
la torre de rectángulo}
```

Informar (cantVertices)
Fin

En este caso, el procedimiento Rectangulo posee dos parámetros de entrada, base y altura, en los cuales recibe los valores de los parámetros actuales ancho y alto del programa principal, respectivamente. Además, el procedimiento Rectangulo, posee un parámetro de entrada/salida, cantidad, utilizado para recibir la cantidad de vértices con flor encontrados hasta el momento e incorporarle la cantidad hallada en este rectángulo. Al terminar al procedimiento, el valor final de cantidad será asignado a cantVertices del programa principal.

Es importante hacer notar que, en las sucesivas invocaciones al procedimiento Rectangulo, como el dato cantVertices se relaciona con el parámetro de entrada/salida cantidad, este dato entra al procedimiento con el valor que indica la cantidad de vértices con flores encontrada hasta el momento. Durante la ejecución de este procedimiento podría modificarse su valor y el efecto se verá reflejado nuevamente en la variable cantVertices del programa principal.

7.3 Otro uso de los parámetros de Entrada/Salida.

Los parámetros de entrada/salida, por permitir la comunicación en ambos sentidos, pueden ser utilizados para reemplazar a los parámetros de entrada (aunque no es recomendable) ó bien para que únicamente retornen valores.

Ejemplo 7.4: Programe al robot para que recorra la calle 10 e informe la cantidad total de esquinas que contienen exactamente 4 papeles.

Para resolver este problema podemos pensar en un procedimiento que cuenta los papeles de una esquina.

```
procedimiento ContarPapeles ( es papeles :numero)
comenzar
  papeles:= 0
  mientras HayPapelEnLaEsquina
    comenzar
      tomarPapel
      papeles := papeles + 1
    fin
  repetir papeles
    depositarPapel
fin
```

Como podemos observar, el procedimiento ContarPapeles recibe el parámetro de entrada/salida papeles y lo primero que hace es inicializarlo en 0 para poder saber cuántos papeles hay en la esquina donde está parado. Al terminar la ejecución del procedimiento el parámetro papeles contiene la cantidad de papeles de esa esquina.

A continuación se presenta la solución completa en DaVinci:

```
programa Cap7Ejemplo4
subprogramas
  procedimiento ContarPapeles( es papeles :numero)
  { definición del procedimiento}

variables
  totalEsquinas4Papeles:numero
  papelesEsquina : numero
comenzar
  iniciar
  Pos (1,10)
  derecha
  totalEsquinas4Papeles := 0 (1)
  repetir 9
    comenzar
    ContarPapeles (papelesEsquina) (2)
    si papelesEsquina = 4 (3)
      totalEsquinas4Papeles := totalEsquinas4Papeles + 1
    mover
  fin
  {análisis de la esquina (100,10)}
  ContarPapeles (papelesEsquina)
  si papelesEsquina = 4
    totalEsquinas4Papeles := totalEsquinas4Papeles + 1
  Informar (totalEsquinas4Papeles)
fin
```

En el punto (1) utilizamos una variable totalEsquinas4Papeles para saber cuántas esquinas tienen exactamente 4 papeles, inicializándola en 0 porque hasta ese momento no hemos contado nada. En (2), se invoca al procedimiento ContarPapeles con el parámetro formal papelesEsquina que no tiene ningún valor, pero como ya vimos, se inicializará en 0, ni bien comience a ejecutarse ese módulo. En el punto (3) se evalúa la cantidad de papeles de la esquina a través de papelesEsquina que devolvió el procedimiento ContarPapeles y si es 4 entonces se actualiza el contador totalEsquinas4Papeles.

Ejemplo 7.5: Programe al robot para que informe la cantidad de calles que contienen más de 50 flores.

Se puede utilizar el módulo desarrollado en el ejemplo 7.1, el código sería el siguiente:

```
programa EntradaSalida
subprogramas
  procedimiento SumarFloresEsquina ( es flores :numero ) (1)
  comenzar
  { este procedimiento fue definido en el ejemplo 7.1 }
  fin

  procedimiento RecorrerCalle (es TotalFlores : numero ) (2)
```

```
comenzar
  TotalFlores := 0
  repetir 9
    comenzar
      SumarFloresEsquina (TotalFlores)
    Mover
  fin
  SumarFloresEsquina (TotalFlores)
fin

variables
  totalCalles50Flores : numero
  floresCalle : numero
comenzar
  iniciar
  derecha
  totalCalles50Flores := 0
  repetir 9
    comenzar
      RecorrerCalle ( floresCalle )
    si floresCalle > 50
      totalCalles50Flores := totalCalles50Flores + 1
      Pos( 1, PosCa + 1)
    fin
  {recorrido de la calle 10}
  RecorrerCalle ( floresCalle )
  si floresCalle > 50
    totalCalles50Flores := totalCalles50Flores + 1
  Informar(totalCalles50Flores)
fin
```

Notemos que en (1) y (2), los parámetros son de entrada/salida. En (1) el módulo utiliza el parámetro flores se utiliza como entrada y salida porque sobre este dato se va acumulando el total de flores de todas las esquinas de una calle. En cambio en (2) el módulo utiliza el parámetro TotalFlores únicamente como salida. Para un uso correcto de este parámetro, el programador no debe olvidar la inicialización al comenzar el procedimiento, porque de lo contrario podrían obtenerse resultados erróneos.

En resumen, un procedimiento que utiliza parámetros de entrada/salida como únicamente de salida es totalmente independiente del módulo que lo invoca. Cuando se usa el parámetro en este sentido, se trata de información generada dentro del procedimiento que se desea dar a conocer al módulo que lo llamó. En este caso no se busca un intercambio de información en ambos sentidos, solo el procedimiento es quien exporta datos.

7.4 Conclusiones

En este apunte se ha buscado introducir algunas ideas útiles a lo largo de la carrera en Informática:

1. Resultan de interés los problemas “solubles por computadora”, es decir expresables como algoritmos.
2. No solo se debe entender como son los problemas, sino aprender a modelizarlos y a resolverlos en forma ordenada y sistemática.
3. No basta con tener UNA solución. Normalmente existen varias. La elegida debe ser EFICIENTE y además la forma en que esté escrita debe ser CLARA y ENTENDIBLE.

Nota: para realizar este apunte se utilizó (con el permiso correspondiente) el material del curso de Ingreso a la Facultad de Informática de la UNLP. En el mismo se han realizado algunos agregados y modificaciones.