Anterior (view.php?id=485) Posterior (view.php?id=775)

12.- Creación de funciones y procedimientos (subprocesos)

Curso: Introducción a la programación - Pseudocódigo (../../course/view.php?id=22)

12 - Creación de funciones y procedimientos (subprocesos)

En muchos casos, nos encontraremos con tareas que tenemos que repetir varias veces en distintos puntos de nuestro programa. Si tecleamos varias veces el mismo fragmento de programa no sólo tardaremos más en escribir: además el programa final resultará menos legible, será más también será más fácil que cometamos algún error alguna de las veces que volvemos a teclear el fragmento repetitivo, o que decidamos hacer una modificación y olvidemos hacerla en alguno de los fragmentos. Por eso, conviene evitar que nuestro programa contenga código repetitivo. Una de las formas de evitarlo es usar "subrutinas", una posibilidad que la mayoría de lenguajes de programación permiten, y que en ocasiones recibe el nombre de "procedimientos" o de "funciones" (existe algún matiz que hace que esas palabras no sean realmente sinónimas y que comentaremos más adelante).

PseInt permite definir "subrutinas" (o "funciones") dentro del pseudocódigo, desde la versión del 10 de octubre de 2012. En su caso, se llaman "subprocesos". Veamos un ejemplo de su uso:

Vamos a empezar por crear un subproceso (o "subrutina", o "procedimiento") que escriba 20 guiones, que podríamos utilizar para subrayar textos. Un programa completo que escribiera tres textos y los subrayara podría ser:

```
Proceso SubProcesos01
    Escribir " Primer ejemplo"
    Para x <- 1 Hasta 20 Hacer
        Escribir Sin Saltar "-"
    FinPara
    Escribir ""

    Escribir " Segundo ejemplo"
    Para x <- 1 Hasta 20 Hacer
        Escribir Sin Saltar "-"
    FinPara
    Escribir ""

    Escribir " Tercer ejemplo"
    Para x <- 1 Hasta 20 Hacer
        Escribir Sin Saltar "-"
    FinPara
    Escribir ""
FinProceso
```

Muy repetitivo. Sería un poco más elegante si lo reescribimos así:

```
Proceso SubProcesos02
    Escribir " Primer ejemplo"
    Subrayar

    Escribir " Segundo ejemplo"
    Subrayar

    Escribir " Tercer ejemplo"
    Subrayar
FinProceso

Subproceso Subrayar
    Para x <- 1 Hasta 20 Hacer
        Escribir Sin Saltar "-"
    FinPara
    Escribir ""
FinSubproceso
```

Mucho más legible, pero todavía no está tan bien como debería: siempre estamos escribiendo 20 guiones, aunque el texto sea más largo o más corto. En la mayoría de lenguajes de programación se puede indicar detalles adicionales ("parámetros") para que se puedan utilizar desde dentro de esa subrutina. Por ejemplo, en nuestro caso podríamos indicarle qué texto queremos escribir y qué longitud queremos que tenga la secuencia de guiones:

```
Proceso SubProcesos03
    EscribirSubrayado(" Primer ejemplo", 16)
    EscribirSubrayado(" Segundo ejemplo", 17)
    EscribirSubrayado(" Tercer ejemplo", 16)
FinProceso

Subproceso EscribirSubrayado(texto, cantidad)
    Escribir texto
    Para x <- 1 Hasta cantidad Hacer
        Escribir Sin Saltar "-"
    FinPara
    Escribir ""
FinSubproceso
```

Eso todavía es un poco redundante: en general, querremos escribir tantos guiones como letras tenga el texto, así que no será necesario indicar ese dato. Desde octubre de 2012, PseInt incluye ciertas funciones predefinidas para manejo de cadenas de texto; una de ellas es "Longitud", que nos indica la cantidad de letras que tiene un texto, de modo que nuestro programa se podría simplificar así:

```
Proceso SubProcesos04
    EscribirSubrayado("Primer ejemplo")
    EscribirSubrayado("Segundo ejemplo")
    EscribirSubrayado("Tercer ejemplo")
FinProceso

Subproceso EscribirSubrayado(texto)
    Escribir texto
    Para x <- 1 Hasta Longitud(texto) Hacer
        Escribir Sin Saltar "-"
    FinPara
    Escribir ""
FinSubproceso
```

También podemos crear subprocesos que realicen ciertas operaciones aritméticas y devuelvan un resultado, como las funciones matemáticas que vimos en el tema 10. Se puede hacer con la misma palabra "subproceso" que hemos empleado hasta ahora, pero muchos lenguajes de programación distinguen entre un "procedimiento" o "subrutina", cuando se da una serie de pasos pero no se devuelve ningún valor (como habíamos hecho hasta ahora), y una "función", cuando sí se devuelve un resultado (como haremos en esta ocasión), así que usaremos esta última nomenclatura. Su desarrollo se parece mucho a lo que hemos hecho hasta ahora, con la diferencia de que escogeremos un nombre para la variable que guardará el resultado, así:

```
Funcion resultado <- Maximo( n1, n2)
    Si n1 > n2  Entonces
        resultado <- n1
    Sino
        resultado <- n2
    FinSi
FinFuncion

Proceso Funciones01
    Escribir "El máximo de 5 y 12 es:"
    Escribir Maximo(5,12);
    Escribir "El máximo de 25 y 12 es:"
    Escribir Maximo(25,12);
Finproceso
```

Este programa crea una función que calcula cuál es el mayor de los dos números que se le indican como parámetro, y la usa dos veces, para mostrar el máximo de dos valores prefijados.

No sólo pdemos devolver valores numéricos; también podemos devolver cadenas (como las funciones que vimos en el apartado 13) o "valores de verdad" (verdadero, falso). Por ejemplo, podemos crear una función que calcule si un número es primo o no (lo vamos a hacer de la forma más simple pero también de la menos eficiente: aplicar la definición, probando a dividir entre todos los números que hay entre 1 y n; si hemos encontrado dos divisores -o menos, para el número uno-, entonces el número es primo):

```
SubProceso resultado <- Primo ( num )
    cantidadDivisores <- 0
    Para cont <- 1 Hasta num Hacer
        Si num % cont = 0 Entonces
            cantidadDivisores <- cantidadDivisores + 1
        FinSi
    Fin Para
    Si cantidadDivisores <= 2 Entonces
        resultado <- verdadero
    Sino
        resultado <- falso
    Fin Si
Fin SubProceso

Proceso PrimosDel1Al30
    Para n <- 1 hasta 30
        si Primo(n) Entonces
            Imprimir n
        FinSi
    FinPara
FinProceso
```

Hay más detalles que comentar sobre funciones, pero son un poco más avanzados, así que vamos a descansar un poco aquí de nuevos conceptos y a practicar lo que hemos visto...

Ejercicio de repaso propuesto 12.1: Crea un procedimiento EscribirCentrado, que reciba como parámetro un texto y lo escriba centrado en pantalla (suponiendo una anchura de 80 columnas; pista: deberás escribir 40 - longitud/2 espacios antes del texto).

Ejercicio de repaso propuesto 12.2: Crea una función CantidadDeDivisores, que reciba un número entero y devuelva la cantidad de divisores (por ejemplo, para el número 16, sus divisores son 1, 2, 4, 8, 16, por lo que la respuesta debería ser 5).

Ejercicio de repaso propuesto 12.3: Crea un programa que pida dos número enteros al usuario y diga si alguno de ellos es múltiplo del otro. Crea una función EsMultiplo que te ayude a que el proceso principal sea legible.

Ejercicio de repaso propuesto 12.4: Crea un procedimiento EscribirEspaciado, que reciba como parámetro un texto y lo escriba con un espacio adicional tras cada letra. Por ejemplo, "Hola, tú" se escribiría "H o l a , t ú ".

Ejercicio de repaso propuesto 12.5: Crea una función MayorDeTres, que reciba tres números enteros y devuelva el valor del mayor de ellos. Por ejemplo, para los números 5, 7 y 5, devolvería el valor 7.

Ejercicio de repaso propuesto 12.6: Crea una función EsPar que devuelva el valor lógico "verdadero" o "falso" según si el número que se indique como parámetro es par o no lo es.

Ejercicio de repaso propuesto 12.7: Crea una función Cubo, que reciba un número y lo devuelva elevado al cubo.

Ejercicio de repaso propuesto 12.8: Crea una función Iniciales, que devuelva una cadena formada por las iniciales de la frase que se indique como parámetro (primera letra y la letra que haya tras cada espacio; por ejemplo, para "Nacho Cabanes" devolvería "NC").

Ejercicio de repaso propuesto 12.9: Crea una función Contiene, que reciba una cadena y una (posible) subcadena, y devuelva "verdadero" o "falso", según si la primera contiene a la segunda (como "Nacho" contiene "ac") o no la contiene, (como "Aurora" no contiene "sol").

Actualizado el: 08-02-2016 15:34

‹ Anterior (view.php?id=485) Posterior › (view.php?id=775)

Novedades

22-03-2015: Puedes votar cuando te guste una respuesta (../../novedades.php)
17-10-2014: Se pueden hacer tests de repaso (../../novedades.php) Anteriores... (../../novedades.php)

- Contactar (../../contactar.php)
- ¿Quiénes somos? (../../quienes.php)