Estructuras cíclicas II

La estructura cíclica para

Jonatan Goméz Perdomo, Ph.D.
jgomezpe@unal.edu.co

Arles Rodríguez, Ph.D.
aerodriguezp@unal.edu.co

Camilo Cubides, Ph.D.(c)
eccubidesg@unal.edu.co

Grupo de investigación en vida artificial – Research Group on Artificial Life – (Alife)
Departamento de Ingeniería de Sistemas e Industrial
Facultad de Ingeniería
Universidad Nacional de Colombia

Agenda

La estructura de control de ciclos para (for)

Existe otra estructura cíclica que es de uso frecuente en programación, el ciclo para (`for`). Esta estructura de control tiene dos propositos primordiales que no siempre son soportados por todo lenguaje de programación:

1. Como una forma compacta de escribir un ciclo mientras (`while`).
2. Para *iterar* sobre los elementos de una colección de elementos.

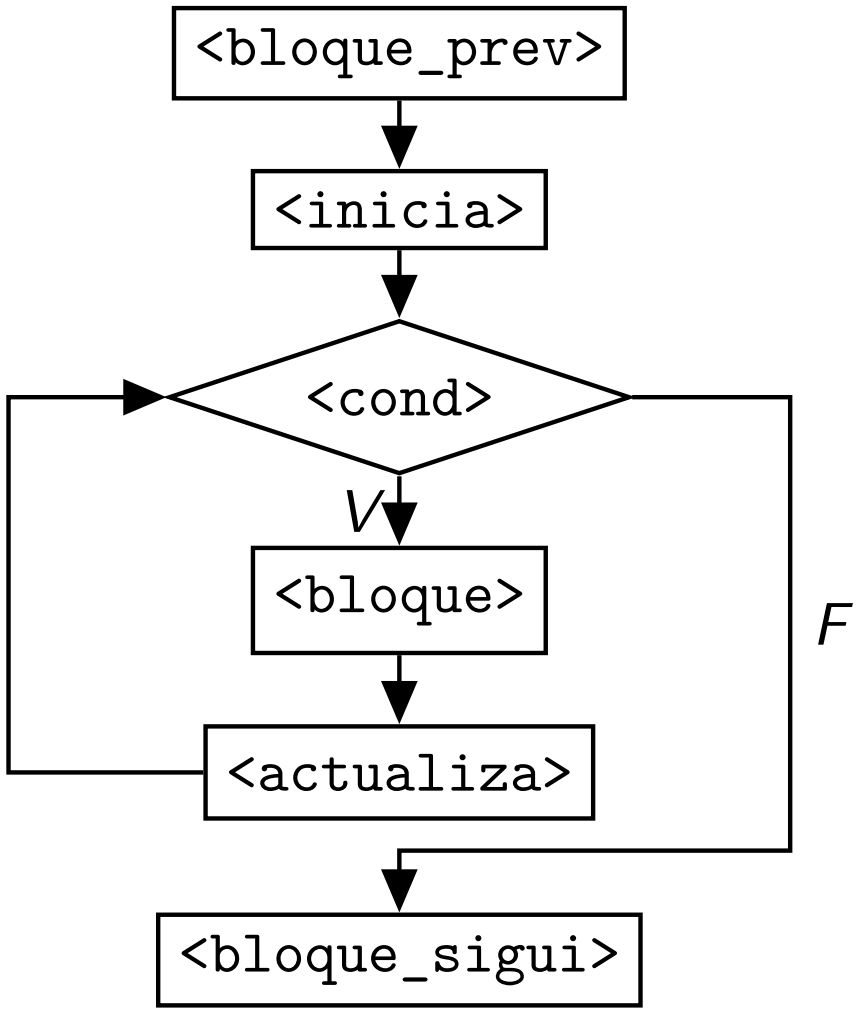
Esta estructura es usualmente utilizada cuando se conocen los valores inicial y final de la variable que es utilizada en la condición de parada.

Agenda

El ciclo para como versión del ciclo mientras I

En muchos lenguajes (incluido Java), el ciclo para (`for`) se usa para escribir de manera compacta un ciclo mientras (`while`). La representación gráfica mediante un diagrama de flujo es la misma que la de un ciclo mientras (`while`), la cual es la siguiente.

El ciclo para como versión del ciclo mientras II



El ciclo para como versión del ciclo mientras III

La sintaxis general de un ciclo mientras (while) en el lenguaje de programación Java es:

```
<bloque_prev>
<inicia>
while(<cond>){
    <bloque>
    <actualiza>
}
<bloque_sigui>
```

El ciclo para como versión del ciclo mientras IV

Un esquema textual que en Java representa un ciclo (for) es la que se da en el siguiente fragmento de código, obsérvese que un ciclo (for) es equivalente a un ciclo (while) pero su representación sintáctica es más compacta y se separa el control del ciclo del calculo que se desea realizar con el ciclo.

```
<bloque_prev>
for(<inicia>; <cond>; <actualiza>){
    <bloque>
}
<bloque_sigui>
```

La suma de los primeros n números naturales I

Ejemplo (La suma de los primeros n números naturales)

Las dos siguientes funciones permiten calcular la suma de los primeros n números naturales positivos, es decir, permiten calcular el valor de la expresión

$$1 + 2 + 3 + \cdots + (n-1) + n, \text{ que abreviadamante se escribe como } \sum_{i=1}^{n} i$$

```
int suma(int n) {
    int s = 0;
    int i = 1;
    while(i <= n){
        s = s + i;
        i++;
    };
    return s;
```

```
public static int suma(int n) {
    int s = 0;
    for (int i = 1; i <= n; i++){
        s = s + i;
    }
    return s;
}
```

La suma de los primeros n números naturales II

Ejemplo (continuación) (La suma de los primeros n números naturales)

estas dos funciones son equivalentes, ya que ejecutan las mismas modificaciones de las variables, pues se tiene que el fragmento de código:

- <bloque_prev> corresponde a la instrucción

```
int s = 0;
```

- <inicia> corresponde a la instrucción

```
int i = 1;
```

- <cond> corresponde a la instrucción

```
i <= n
```

La suma de los primeros n números naturales III

Ejemplo (continuación) (La suma de los primeros n números naturales)

- <bloque> corresponde a la instrucción

```
s = s + i;
```

- <actualiza> corresponde a la instrucción

```
i++;
```

- <bloque_sigui> corresponde a la instrucción

```
return s;
```

La suma de los primeros n números naturales IV

Ejemplo (continuación) (La suma de los primeros n números naturales)

En la construcción de estas funciones aparecen dos variable que tienen una connotación muy importante:

- La variable `i` juega el rol de una *variable contadora* ya que ésta permite llevar el conteo de cuantos ciclos se han efectuado.
- La variable `s` juega el rol de una *variable acumuladora* pues en ésta se acumula o almacena el valor parcial que se desea calcular utilizando el ciclo.

La suma de los primeros n números naturales V

Ejemplo (continuación) (La suma de los primeros n números naturales)

La codificación en Java de una función que permite sumar los primeros n números naturales positivos junto con su programa principal es

```
import java.util.Scanner;
public class Sumatoria {
```

```
public static int suma(int n) {
    int s = 0;
    for (int i = 1; i <= n; i++) {
        s = s + i;
    }
    return s;
}
```

La suma de los primeros n números naturales VI

Ejemplo (continuación) (La suma de los primeros n números naturales)

```
    public static void main(String[] args) {
        int n;
        Scanner sc = new Scanner(System.in);
        System.out.print("n? ");
        n = sc.nextInt();
        System.out.println("La suma de los primeros "
        + n + " numeros naturales es:");
        System.out.println(suma(n));
    }
}
```

La suma de los primeros n números naturales VI

Ejemplo (continuación) (La suma de los primeros n números naturales)

Si se ejecuta el anterior programa y como entrada se ingresa el valor `n` $= 6$ (por ejemplo el número de caras de un dado ⚀ + ⚁ + ⚂ + ⚃ + ⚄ + ⚅), el resultado que se obtiene es el siguiente

```
n? = 6
La suma de los primeros n números es: 21
```

La suma de los primeros n números naturales VII

Ejemplo (continuación) (La suma de los primeros n números naturales)

En general es fácil comprobar si el resultado que se obtiene utilizando la función `suma(n)` es correcto, pues existe una fórmula muy sencilla para calcular la suma de los primeros n números naturales positivos sin necesidad de realizar la suma exhaustiva. Esta fórmula es

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

Agenda

Los ciclos para (for) I

Pueden existir mas de una expresion de inicio asi como tambien mas de una expresion de actualización

```
for(int i=0, j=10; i<=j; i++, j--){
    System.out.println(i+","+j);
}
```

```
0, 10
1, 9
2, 8
3, 7
4, 6
5, 5
```

Los ciclos para (for) II

Es necesario tener muy en claro el comportamiento del ciclo, ya que una condicion o una expresion de actualización mal diseñada podrian dar origen a un ciclo infinito

```
for(int i=1; i<=14; i--){
    System.out.println(i);
}
```

```
 1
 0
-1
-2
-3
 ...
```

La variable i nunca llega a ser 14, de modo que el ciclo sigue ejecutandose indefinidamente.

Finalizando un ciclo (for)

Tal y como se mencionó anteriormente, la expresion break resulta en la terminación forzada del ciclo, sin importar cuál sea la condición de terminación del ciclo.

Ejemplo (Finalizando un ciclo for)

```
for(int i=0; i<=30; i++){
    if(i==4){
        break;
    }
    System.out.println(i);
}
```

```
0
1
2
3
```

Problemas varios I

Problemas

1. Imprimir un listado con los números del 1 al 100 cada uno con su respectivo cuadrado.
2. Imprimir un listado con los números impares desde 1 hasta 999 y seguidamente otro listado con los números pares desde 2 hasta 1000.
3. Imprimir los números pares en forma descendente hasta 2 que son menores o iguales a un número natural $n \geq 2$ dado.
4. Imprimir los números de 1 hasta un número natural n dado, cada uno con su respectivo factorial.
5. Calcular el valor de 2 elevado a la potencia n.
6. Leer un número natural n, leer otro dato de tipo real x y calcular x^n.
7. Diseñe un programa que muestre las tablas de multiplicar del 1 al 9.

Problemas varios II

Problemas

8. Diseñar una función que permita calcular una aproximación de la función exponencial alrededor de 0 para cualquier valor $x \in \mathbb{R}$, utilizando los primeros n términos de la serie de Maclaurin

$$\exp(x, n) \approx \sum_{i=0}^{n} \frac{x^i}{i!}.$$

9. Diseñar una función que permita calcular una aproximación de la función seno alrededor de 0 para cualquier valor $x \in \mathbb{R}$ (x dado en radianes), utilizando los primeros n términos de la serie de Maclaurin

$$\sin(x, n) \approx \sum_{i=0}^{n} \frac{(-1)^i x^{2i+1}}{(2i+1)!}.$$

Problemas varios III

Problemas

10. Diseñar una función que permita calcular una aproximación de la función coseno alrededor de 0 para cualquier valor $x \in \mathbb{R}$ (x dado en radianes), utilizando los primeros n términos de la serie de Maclaurin

$$\cos(x, n) \approx \sum_{i=0}^{n} \frac{(-1)^i x^{2i}}{(2i)!}.$$

11. Diseñar una función que permita calcular una aproximación de la función logaritmo natural alrededor de 0 para cualquier valor $x \in \mathbb{R}^+$, utilizando los primeros n términos de la serie de Maclaurin

$$\ln(x, n) \approx \sum_{i=0}^{n} \frac{1}{2i+1} \left(\frac{x^2 - 1}{x^2 + 1} \right)^{2i+1}.$$

Problemas varios III

Problemas

12. Diseñar una función que permita calcular una aproximación de la función arco tangente para cualquier valor $x \in [-1, 1]$, utilizando los primeros n términos de la serie de Maclaurin (al evaluar esta función el resultado que se obtiene está expresado en radianes)

$$\arctan(x, n) \approx \sum_{i=0}^{n} \frac{(-1)^i x^{2i+1}}{(2i+1)}.$$