

Tema 4. Esquemas Condicionales

Objetivos

- **Conocer los tipos de esquemas condicionales**
- **Entender su funcionamiento y sus diferencias**
- **Ser capaz de aplicarlos en ejemplos sencillos**

4. Esquemas Condicionales

1. Introducción

1. **Los esquemas condicionales en los esquemas de composición de acciones**

2. Esquemas condicionales

1. **Sintaxis**
2. **Semántica**
3. **Ejemplos**

3. Esquemas condicionales en C

1. **Sintaxis**
2. **Ejemplos**

4.1. Introducción

4.1.1. Los esquemas condicionales en los esquemas de composición de acciones

Ya hemos visto el Concepto de programación estructurada (Tema 2. Bloque 2) (Bohm y Jaccopini, 66). Los programas **proprios** se pueden realizar **exclusivamente** mediante 3 esquemas:

- Secuencial (tema 2),
- **Condicional (tema 4) e**
- Iterativo (tema 5)

<acción> ::= <acción-elemental> | <algoritmo> |
<composición-secuencial-de-acciones> |
<esquema-condicional> |
<esquema-repetitivo>

4.2. Esquemas Condicionales

- **Ejecución de una acción (o conjunto de acciones) en función del entorno.**
- **Si se cumple una condición en el entorno, se realiza un conjunto de acciones.**
- **Evaluar un condicional, equivale a evaluar una expresión lógica: simple o compuesta (Tema 2).**
- **Si no se cumple dicha condición en el entorno, se realiza otro conjunto de acciones (que puede ser vacío).**

4.2.1. Sintaxis

<esquema-condicional> ::=
 <esquema-condicional-1> |
 <esquema- condicional-2> |
 <esquema- condicional-generalizado>

<esquema-condicional-1> ::=
 si <expresión-lógica>
 entonces <acción-1>
 fin si

<esquema-condicional-2> ::=
 si <expresión-lógica>
 entonces <acción-1>
 sino <acción-2>
 fin si

4.2.2. Semántica

Esquema condicional 1: *si_entonces_fin_si*

- Se ejecuta la <acción-1> si la expresión lógica se evalúa a cierto. Después se ejecuta la acción siguiente al esquema *si_entonces_fin_si*.
- Si se evalúa a falso, no hace nada y pasa a la acción siguiente al esquema *si_entonces_fin_si*.

Esquema condicional 2: *si_entonces_sino_fin_si*

- Se ejecuta la <acción-1> si la expresión lógica se evalúa a cierto. Después se ejecuta la acción siguiente al esquema *si_entonces_sino_fin_si*.
- Si se evalúa a falso, se ejecuta la <acción-2> y se pasa a la acción siguiente al esquema *si_entonces_fin_si*.

4.2.3. Ejemplos de esquemas condicionales

1. Cálculo del valor absoluto de un número:

Resumen: Dado un número, calculamos su valor absoluto

Entorno: 2 variables numéricas:

X: numérico; {dato}

Y: numérico; {resultado}

P.B.R. Si $X < 0$ entonces Y será $-X$; Si no, Y será X

2. Crea un programa que dado un número entero entre 1 y 7, indique qué día de la semana corresponde (como cadena de texto)

Resumen: Dado un número entre 1 y 7, decir qué día de la semana representa.

Entorno:

Día: Numérico; {p. dato}

Letra: Cadena; {p. resultado}

Principio básico de resolución: Comparo el número que me dan con los que están en el rango 1..7. Si no está en ese rango, se escribe un mensaje de error, y la Letra será "Error"

Programa principal

Algoritmo Principal es

A, B: numérico;

Inicio

 Escribir "Dame un número: ";

 Leer A;

 absoluto (A, B);

 Escribir B, " es el valor absoluto de ", A;

Fin

{Versión 1}

Algoritmo Absoluto (X, Y) es

X: numérico; {par. Dato}

Y: numérico; {par. Resultado}

Inicio

 Y := X;

 si X < 0 entonces

 Y := -Y;

 fin si

Fin

{Versión 2}

Algoritmo Absoluto (X, Y) es

X: numérico; {par. Dato}

Y: numérico; {par. Resultado}

Inicio

 si X < 0 entonces

 Y := -X;

 sino

 Y := X;

 fin si

Fin

4.2.1. Sintaxis Esquema Condicional Generalizado (cont.)

<esquema- condicional-generalizado> ::=

según <expresión> **hacer**

 <caso>

 {; <caso>}

 [; <otro-caso>]

fin según

<caso> ::= <valor> {, <valor>} : <acción>

<otro-caso> ::= otros: <acción>

- <expresión> debe tomar valores discretos (enteros, carácter o lógicos)

SEMÁNTICA

- Los valores de cada caso deben ser mutuamente excluyentes
- Se realizará la <acción> asociada **uno y sólo uno** de los casos
- Tras realizar una <acción> termina, yendo el control a la siguiente expresión al **fin según**

Algoritmo Dia_Semana (Día, Letra) **es**

Día: Numérico; {par. dato}

Letra: Cadena; {par. resultado}

Inicio

si Día = 1 entonces

 Letra := "Lunes"

sino {Día no es lunes}

 si Día = 2 entonces

 Letra := "Martes"

 sino {Ni lunes ni martes}

 si Día = 3 entonces

 Letra := "Miércoles"

 sino {Ni lunes, ni martes ni miércoles.}

 si Día = 4 entonces

 Letra := "Jueves"

 sino {ni lunes, ni martes, ni miércoles ni jueves}

 si Día = 5 entonces

 Letra := "Viernes"

 sino {ni lunes, ni martes, ni miércoles,
 ni jueves, ni viernes}

 si Día = 6 entonces

 Letra := "Sábado"

 sino {ni lunes, ni martes, ni miércoles,
 ni jueves, ni viernes ni sábado}

 si Día = 7 entonces

 Letra := "Domingo"

 sino {Error}

 Letra := "Error";

 escribir "Error";

 finsi {Día = 7}

 finsi {Día = 6}

 finsi {Día = 5}

 finsi {Día = 4}

 finsi {Día = 3}

 finsi {Día = 2}

 finsi {Día = 1}

Fin

Ejemplos esquema condicional generalizado (cont.)

Ejemplo: calcular día de la semana en función del número de día de la semana

Algoritmo Dia_Semana (Día, Letra) es

Día: Numérico; {par. dato}

Letra: Cadena; {par. resultado}

Inicio

según Día hacer

1: Letra := "Lunes"; {Día = 1}

2: Letra := "Martes"; {Día = 2}

3: Letra := "Miércoles"; {Día = 3}

4: Letra := "Jueves"; {Día = 4}

5: Letra := "Viernes"; {Día = 5}

6: Letra := "Sábado"; {Día = 6}

7: Letra := "Domingo"; {Día = 7}

otros: Letra := "Error"; escribir "Error";

fin según

Fin

4.3. Esquemas condicionales en C

- **CONDICIONAL-1 (SIMPLE)**
if (<condición != 0>) <sentencia_1>
if (<condición != 0>) { <composición_secuencial> }
- **CONDICIONAL-2 (COMPUESTO)**
if (<condición != 0>) {
 <sentencias_cierto>
}
else {
 <sentencias_falso>
}
- **CONDICIONAL GENERALIZADO**
switch (<expresión_escalár>) {
 {case <cte.1>: <sentencias>; [break;]}
 default: <sentencias>;
}

Recapitulación sobre esquemas condicionales 1 y 2

¿Cómo interpretar estructuras condicionales anidadas?

- 1. El término entonces está asociado al término si más próximo, que no tenga ya asociado un entonces.**
- 2. El delimitador sino está asociado al término si más próximo, que no tenga ya asociado un sino.**
- 3. El delimitador finis está asociado al término si más próximo, que no tenga ya asociado un finis.**