INSTITUTO KHIPU

Semestre I

Sesión 25

CARRERA PROFESIONAL

DESARROLLO DE SISTEMAS DE INFORMACIÓN

FUNDAMENTOS DE PROGRAMACIÓN

Tema

ESTRUCTURA DE CONTROL MULTIPLE

Conceptos Basicos

Ahora vamos a conocer sobre las estructuras de control multiple(Según. Sea, caso de/case).

Con frecuencia en la práctica es necesario que existan más de dos elecciones posibles (por ejemplo, en la resolución de la ecuación de segundo grado existen tres posibles alternativas o caminos a seguir, según que el discriminante sea negativo, nulo o positivo). Este problema, como se verá más adelante, se podría resolver por estructuras alternativas simples o dobles, anidadas o en cascada; sin embargo, este método si el número de alternativas es grande puede plantear serios problemas de escritura del algoritmo y naturalmente de legibilidad[1].

La estructura de decisión múltiple evaluará una expresión que podrá tomar n valores distintos, 1, 2, 3, 4, ..., n. Según que elija uno de estos valores en la condición, se realizará una de las n acciones, o lo que es igual, el flujo del algoritmo seguirá un determinado camino entre los n posibles.

Los diferentes modelos de pseudocódigo de la estructura de decisión múltiple se representan de la siguiente forma.

Sentencia switch (C , C++, Java, C#)

```
switch (expresión) {
        case valor1:
                sentencia1;
                sentencia2;
                sentencia3;
```

[1] http://biblioteca.univalle.edu.ni/files/original/ec89128132c7434f1765ceb9cbf1160828ae85f5.pdf

```
    break;
    case valor2:
            sentencia1;
            sentencia2;
            sentencia3;
    break;
    default:
     sentencia1;
     sentencia2;
     sentencia3;
    // fin de la sentencia compuesta
```

Algunos modelos par usar la estructura de control multiple

Modelo 1:

```
según_sea expresion (E) hacer
     e1: accion S11
         accion S12
         .
         .
         accion S1a
     e2: accion S21
         accion S22
         .
         .
         accion S2b
     .
     .
     en: accion S31
         accion S32
         .
         .
         accion S3p
     si-no
         accion Sx
fin_según
```

Modelo 2 (simplificado):

```
según E hacer
        .
        .
        .
fin_según
```

Modelo 3 (simplificado):

```
opción E de
  .
  .
fin_opción
```

Modelo 4 (simplificado):

```
caso_de E hacer
  .
  .
  .
fin_caso
```

Modelo 5 (simplificado):

```
si E es n hacer
  .
  .
  .
fin_si
```

Diagrama de flujo

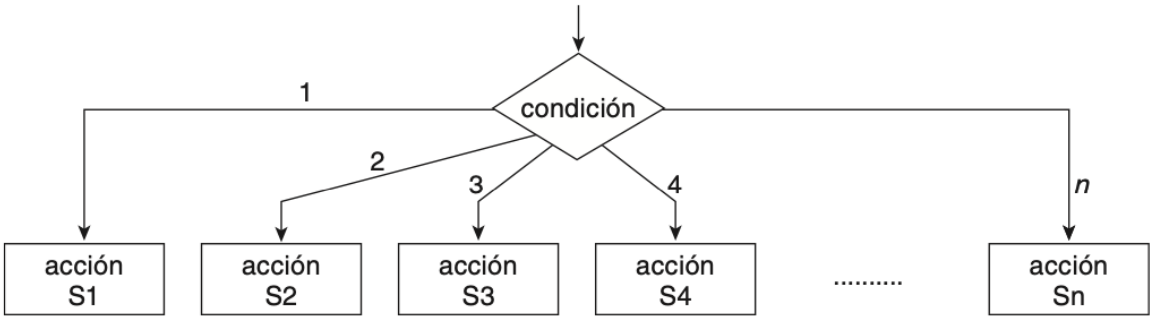
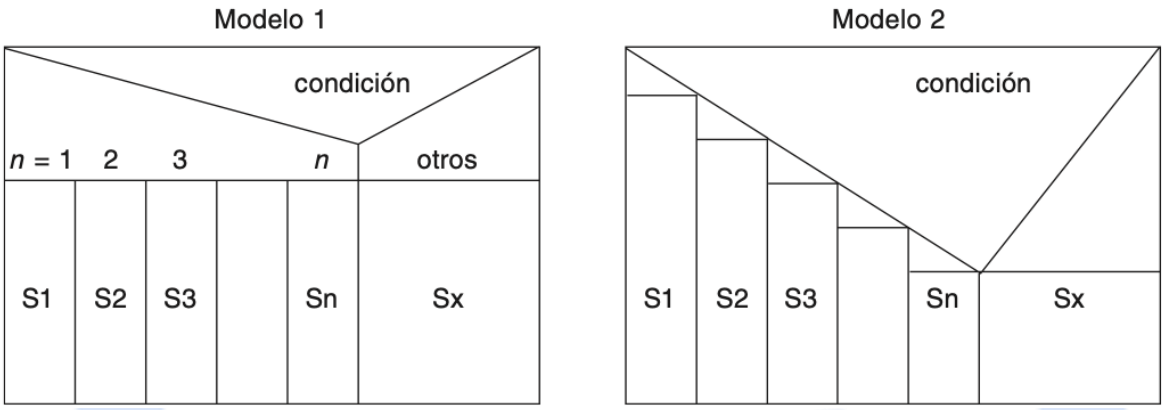


Diagrama N-S



Pseudocódigo

En ingles la estructura de decisión múltiple se representa

case *expresión* **of**

[e1]: *acción S1*

[e2]: *acción S2* .

.

[en]: *acción Sn*

otherwise

acción Sx

end_case

¿Cuál de los modelos expuestos se puede considerar representativo? En realidad, como el pseudocódigo es un lenguaje algorítmico universal, cualquiera de los modelos se podría ajustar a su presentación; sin embargo, nosotros consideramos como más estándar los modelos 1, 2 y 4. En esta obra seguiremos normalmente el modelo 1, aunque en ocasiones, y para familiarizar al lector en su uso, podremos utilizar los modelos citados 2 y 4.

Los lenguajes como C y sus derivados C++, Java o C# utilizan como sentencia selectiva múltiple la sentencia switch, cuyo formato es muy parecido al modelo 6.

Ejemplo

Se desea diseñar un algoritmo que escriba los nombres de los días de la semana en función del valor de una varia- ble DIA introducida por teclado.

Losdíasdelasemanason7;porconsiguiente,elrangodevaloresdeDIAserá1 .. 7,ycasodequeDIAtome un valor fuera de este rango se deberá producir un mensaje de error advirtiendo la situación anómala.

algoritmo DiasSemana
var
entero: DIA
inicio
leer(DIA)
según_sea DIA **hacer**
1: **escribir**('LUNES')
2: **escribir**('MARTES')
3: **escribir**('MIERCOLES') 4: **escribir**('JUEVES')
5: **escribir**('VIERNES') 6: **escribir**('SABADO')
7: **escribir**('DOMINGO') **sí-no**

```
        escribir('ERROR')

        fin_según

fin
```

INSTITUTO

KHIPU