Clase 8-Estructuras de control

domingo, 2 de junio de 2024 11:31

Medio por el cual se determina el flujo de ejecución entre los componentes del programa.

A nivel de Unidad: Cuando el flujo de control pasa entre unidades. Pasajes de parámetros

A nivel de sentencia:

- **Secuencia**: Ejecución de una después de la otra. Algunos lenguajes usan ; para delimitar instrucciones, otras una instrucción por línea y punto. Puedo hacer sentencias compuestas (agruparlas) con begin-end , {}, etc.
  - Asignación: cambia datos de memoria: asigna al l valor el r valor de una expresión. Modifico valor de memoria a partir del resultadao de una expresión. Tiene efectos colaterales
  - Expresión: unidad de código que me devuelve un resultado.

```
(x + 3)*2          // expresión
a > b && c < d     // expresión
y = (x + 3)*2;     // sentencia
```

  - **Está permitido:**
    - a=b=c=0; (En C se evalúa de **derecha a izquierda)**
    - if (i=30) printf("Es verdadero") asigna y evalúa
  - **No está permitido:**
    - ++ p = *q;
    - (i<j?z:y)=4;
    - La ***mayoría de los lenguajes*** de programación **requieren** que sobre el ***lado izquierdo de la asignación aparezca un l-valor y no un r-valor***
- **Selección**: If, ejecutar cosas según si se da o no una cosa.
  - Fortran- If lógico:
    If(condición) sentencia-> 1 camino posible
  - ALGOL if-then-else -> 2 caminos posibles
    AMBIGÜEDAD(algol) If then else anidados->: no se establecía la asociación de los else con los ifs abiertos
    SIN AMBIGÜEDAD(PL/1, Pascal y C) -> cada rama else cierra el último if abiertro (hacia atras) -> copado pero difícil de leer
    - Para dar claridad, se ,ete sentencia de fin del cierre
      por ejemplo
      - **fi** en **Algol 68**
      - **end if** en **Ada**
      - **end Modula-2**
    - Otra solución para lo de la claridad es usar un begin end y adentro el if (ALGOL 60)
  - C if then else -> (No usa then, a diferencia de pascal lleva () en la condición

```
If (condición)  Instrucción 1;
else Instrucción A;
```

  - C if corto (cond ? b: c)
  - Python if then else-> no es ambiguo y es legibole (incorpora elif si hay más de 2 opciones y sangría). Se usan : desdpués del if, else, elif. Y la indentación es obligatoria. El () en la condición es opcional.
  - Python if corto A if C else B

C ? A : B (en C)

A if C else B (en Python)

CORTO CIRCUITO Y LARGO CIRCUITO

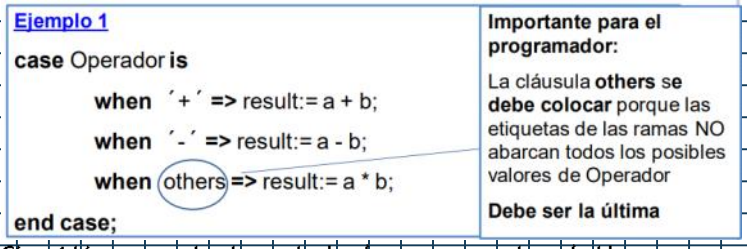
Tiene que ver con cómo se evaluan las expresiones booleanas que tienen ands, or y eso
Corto: si en un and el primero es falso, corta ahí. Si en un or es verdadero, corta. -> optimiza rendimiento, evita errores (pq capaz la segunda condición tiene un valor nulo y entonces no lo

analizo)
El largo sigue la evaluación. a and false and f() en el corto corta si da false y no se hace F. En el largo se corre F igual

- Selección multiple: CASE. No importa el orden en el que aparecen, puede haber un else (pero es opcional) se termina con un end
  Es inseguro porque no establece qué pasa cuando un valor no entra en una de las alternativas
- En Pascal ->

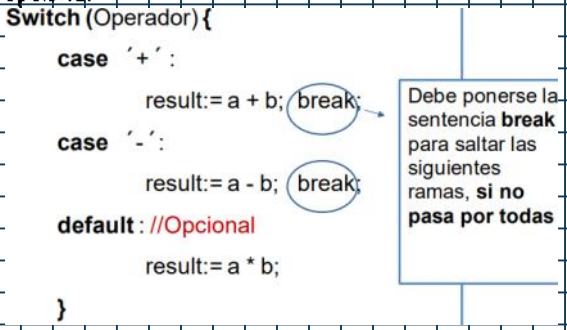
```
Ejemplo:
var opcion : char;
begin
 readln(opcion);
case opcion of
     '1' : nuevaEntrada;
     '2' : cambiarDatos;
     '3' : borrarEntrada
     else
      writeln('Opcion no valida!!')
     end;
end
```

- ADA: Case expresión is con when y end case. Expresiones son solo enteras o enumerativas. Se usa una flechita para indicar lo que se hace si se cumple la condición. Usa others para representar los valores que no entran. Others tiene que estar antes del end.



Sí o sí tienen que tratarse todos los casos o meter el others

- C, C++ usa un switch y cada condición es entero o char (constante). Se puede usar un break para salir de la rama. Si no lo pongo, ejecuta también lo de las ramas de abajo. Existe el default y es opcional



- Ruby -> Case seguido de when y end. La expresión es cualquier cosa que de resultado. Ej 1==1. el else es opcional
- PL/I: WHEN / OTHERWISE / END

```
----------Formato - 1----------
SELECT (identifier) ;
        WHEN (value1) statement;
        WHEN (value2) statement;
        OTHERWISE statement ;
END;

----------Formato - 2 ----------
SELECT ;
        WHEN (cond1) statement;
        WHEN (cond2) statement;
        OTHERWISE statement ;
END;
```

- Iteración: Ejecutar algo muchas veces (si no sabía esto en el 3er año de la carrera estamos en problemas xd)