Sintaxis en C y Java

Vazquez Rocha Jorge Ivan

3 de febrero de 2020

1. Lenguaje C

1.1. Declaración de tipos de datos básicos o primitivos

Una declaración de variables, asocia los identificadores definidos por el usuario con los tipos predefinidos C o bien con los tipo definidos por el usuario.
Mediante una declaración de variable, se reserva una locación de memoria para almacenar datos del tipo definido.

Sintaxis: **Tipo** Variable1, Variable2, . . . ; Dónde:

Tipo es la palabra reservada que denota el tipo al cual pertenece la lista de variables,
Variable1, etc. son los nombres de las variables definidas por el usuario.

Ejemplo:

long double X, Y, Z;
int I, J, K;
enum Sex { Masculino, Femenino };
float Arreglo[10];
struct Ficha { char Nombre[15], Cedula[12];
Sex Sexo;
}Fichero[15];

Nota: toda variable declarada a este nivel tiene carácter global.

1.2. Controles de fujo, condicionales y ciclos

1.2.1. While

Repite un conjunto de instrucciones un número indeterminado de veces mientras una condición se cumpla.

Sintaxis: **while** (Condición) {Instrucciones;} Donde:

-Instrucción representa el conjunto de instrucciones que se desean repetir
-Condición es cualquier expresión lógica, que puede involucrar en su interior cualquiera de los otros tipos de expresiones.

La condición de parada se evala antes de ejecutar cualquier instrucción dentro del **while** en cada iteración, así, si la expresión es falsa el flujo del programa salta hasta la instruccin que se encuentra inmediatamente después de la última instruccin del **while**.

1.2.2. Do while

Al igual que la instrucción anterior, repite un conjunto de instrucciones un número indeterminado de veces mientras una condición se cumpla, con la diferencia que al usar **do while**, se garantiza que el bloque de instrucciones contenidos en la instrucción, se ejecuten al menos una vez, pues la condición se evalúa al final y no al principio como ocurre con **while**.

Sintaxis: **do** { Instrucciones;}
while (Condición);

1.2.3. For

La instrucción **for** es una de las instrucciones más poderosas que presente el lenguaje, aunque su funcionamiento es similar a la instrucción **while**.

sintaxis: **for**(Ininicializaciones; Expresión de Control; Actualizaciones)
{ Instructiones;} Donde:

las secciones entre paréntesis separadas por los símbolos (;) denotan las distintas partes de la sentencia:

- **Ininicializaciones:** En este bloque pueden colocarse instrucciones que se realizaran sólo al principio de la instrucción y suele usarse para inicializar variables.

- **Expresión de Control:** Es una expresión lógica que se evalúa al inicio de cada ciclo y de resultar falsa se continua en la instrucción ubicada después de la llave que denota la finalización del bucle; pero de cer cierta se procede a ejecutar el cuerpo del **for** que son las Instrucciones ubicadas entre las llaves.

- **Actualizaciones:** Son instrucciones que se ejecutan al final de cada iteración, justo antes de evaluar la Expresión de Control y determinar si se realizará un nuevo ciclo.

1.2.4. If, else

Con este tipo de instrucción, se especifican las condiciones bajo las cuales una o un grupo de instrucciones pueden ser ejecutadas.

Sintaxis: **if**(Condición){ Instrucciones; }
else { Instrucciones;} Donde:

Las llaves {} enmarcan el bloque de Instrucciones que se deben ejecutar si se cumple la condición especificada despues del comando **if**, y sólo son necesarios si se espera que se ejecute mas de una instrucción.

Else (de lo contrario) denota el bloque de instrucciones que deben ejecutarse de no cumplirse la condición.

1.2.5. Switch

La instrucción switch, consta de una expresión que es evaluada y según su resultado o valor, se ejecuta uno u otro bloque de instrucciones, la instrucción a ser evaluada debe ser entera.

*Sintaxis:***switch** (Expresión){
case expresion constante 1 : Instrucciones; [break;]
case expresion constante n : Instrucciones; [break;]
default: Instrucciones; } Donde:

Switch, case, default, son palabras reservadas que forman parte de la instrucción.

La sentencia **switch** funciona de la siguiente forma:

La expresión es evaluada, luego se compara con cada expresión constante ubicada al lado de un **case** y en caso de resultar iguales se procede a ejecutar el bloque de instrucciones correspondientes.

Para asegurar que una vez que se encuentra una expresión constante concurrente, las demás no sean evaluadas, se suele escribir la palabra reservada **break** al final del bloque de instrucciones.

Si ninguna de las expresiones constantes satisface la condición de igualdad, se ejecutan las instrucciones correspondientes a la palabra reservada **default**.