COMPILADORES 1

HASHTAG

MANUAL DE USUARIO

ING. CARLOS VALLEJO

INTEGRANTES:

STEPHANIE SCHOENHERR 1094-1150

WILMER CARRANZA 1101-1227

24 DE MARZO DE 2015

Índice

INTRODUCCIÓN HASHTAG

Hashtag es un lenguaje de programación sencillo y amigable al usuario que presenta características similares a Java. La sintáxis que define a este lenguaje permite al desarrollador crear libremente su código para formar un programa.

Los elementos que contiene son los siguientes:

- Identificadores
- Tipos de Datos
- Palabras Reservadas
- Comentarios
- Operadores
- Expresiones
- Bloques de Código

PROGRAMANDO EN HASHTAG

Cada programa debe comenzar con la palabra reservada "begin" y terminar en "end" dentro de ello se creará el programa deseado, como la creación de métodos, además de esto cuenta con la función principal llamada "mainbegin" "do" y termina con "end". Dentro de esa función se permitirá la declaración e inicialización de variables, estructuras de control, instrucciones y llamados. En Hashtag los nombres son sensibles a mayúsculas y minúsculas, es muy importante el nombre de las palabras reservadas todas deben ser minúsculas.

IDENTIFICADORES

Un identificador es un nombre que contiene un valor que puede cambiar a lo largo del programa. Estas pueden ser de tipo entero, punto flotante, carácter, cadena o booleano.

Los nombres de los identificadores se pueden crear a criterio del desarrollador siguiendo el patrón definido. Un conjunto de caracteres alfanuméricos, comenzando siempre con una letra ya sea mayúscula o minúscula y puede incluir el símbolo "_".

TIPOS DE DATOS

Hashtag dispone de 4 tipos de identificadores.

- Boolean: Tipos que almacena únicamente los valores *True* y *False*. EL resultado de la expresión lógica que aparece como condición en un bloque de decisión debe ser boolean.
- Int: Tipos que almacenan valores numéricos (Enteros).
- Double: Tipos que almacenan valores numéricos (Reales).
- Char: Tipos que almacena caracteres.
- String: Tipos que almacena un conjunto de caracteres.

DEFINICIÓN E INICIALIZACIÓN DE IDENTIFICADORES

Definición de un identificador

Un identificador se especifica con el tipo y nombre. Si no se especifica el tipo del identificador creado, este muestra un error. El valor puede o no puede ser especificado al momento de la creación.

Tipo NombreIdentificador;

Tipo NombreIdentificador = Valor;

Tipo NombreIdentificador = Expresión;

Ejemplos de declaración e inicialización de variables:

```
int Var1;                {Declaración sin inicializar}
int Var2=10;             {Declaración inicializada con valor igual a 10}

double x;                {Declaración sin inicializar}
double y=2.5;            {Declaración inicializada con valor 2.5}

boolean bl=true;         {Declaración inicializada con valor true}
boolean b2=false;        {Declaración inicializada con valor false}

char caracter='a';       {Declaración de un caracter}

string cadena_Vacia=""; {Declaración de una cadena vacia}
string cadena="Hola mundo"; {Declaración de una cadena Hola mundo}
```

En el ejemplo se distingue las distintas formas que se puede nombrar un identificador, así como ver a qué tipo pertenece y que valor contiene. Estos identificadores ahora podrán ser utilizados por el desarrollador en el caso que lo necesite.

Ejemplo #2

```
1
2   #Creacion Multiple Identificadores
3   int x, z, y;
4   double var1, var2;
5   string cad,mensaje,ss2;
6   #Tambien se pueden declarar un identificador e inicializarla con una expresión
7   int x = (a + b);
8
9   int numero = 8*10;
10
```

PALABRAS RESERVADAS

Estas palabras reservadas son las que emplea el lenguaje Hashtag y no pueden ser utilizadas como identificadores. Estas palabras reservadas tienen un significado específico para el compilador cada una tiene su función con el cual fue definido por la gramática del lenguaje. A continuación se muestra la lista de palabras reservadas correspondientes a Hashtag.

Lista palabras reservadas

boolean	**if**	**other**	**readint**
break	**int**	**end**	**readdouble**
case	**public**	**mainbegin**	**readchar**
char	**return**	**end**	**readstring**
do	**string**	**and**	**void**
double	**switch**	**or**	
else	**true**	**print**	
false	**while**	**function**	
for	**not**	**begin**	

OPERADORES

Hashtag cuenta con:

- **Operadores aritméticos** como suma (+), resta (-), multiplicación (*), división (/), y resto de la división (%).
- **Operador de asignación** el cual permite asignar un valor a una variable (=).
- **Operadores relacionales** sirven para realizar comparaciones de igualdad, desigualdad así como relación de menor o mayor. (<,>,<=,>=,!=,==).
- **Operadores Lógicos** estos se utilizan para la construcción de expresiones lógicas (true, false, and , or , not).
- **Operadores Incrementales** (++) incrementa en una unidad al identificador que se le aplica.
- **Operadores Decrementales** (--) reduce en una unidad al identificador que se le aplica.
- **Operador de Asignación** (=) permite asignar valores a un identificador.
- A continuación un ejemplo de cómo utilizar los operadores en Hashtag.

```
#Operadores Hashtag

    double resultado, num1, num2=10;

    resultado=num1*num2;
    resultado=num1+num2;
    resultado=num1-num2;
    resultado=num1/num2;
    resultado=num1%num2;

    boolean mayorDeEdad, menorDeEdad;
        int edad = 21;
         mayorDeEdad = edad >= 18;   #mayorDeEdad será true
         menorDeEdad = not mayorDeEdad; #menorDeEdad será false


    boolean fuego =true, agua=false, granizo=false;
    boolean condicion= fuego or agua or graniz;
```

```
1
2
3   int x = 5;     #Identificadore de tipo entero inicializado con el valor igual a 5
4   int y = 5;
5   int z;
6   z =  x++;      #operador de incremento, incrementa en una unidad.
7   z =  Y--;      #operador de decremento, decrementa en una unidad.
8
9
10
```

COMENTARIOS

Hashtag cuenta con dos formas diferentes de introducir comentarios. Los comentarios le sirven al desarrollador para tener una documentación del código y entenderlo de una forma más rápida, facilitando las correcciones y revisiones del mismo. Realizar comentarios debe ser una buena práctica ya que otros desarrolladores lograrían entender mejor la funcionalidad y poder contribuir.

Un comentario en una línea se puede colocar en cualquier parte del código. De la siguiente manera:

 # Este es un comentario en una línea.

Un comentario de múltiples líneas comienza con una llave izquierda seguido del texto y termina con una llave Derecha, se escribe de la siguiente manera:

{Comentario de multiples líneas

Ejemplo escrito en lenguaje Hashtag.}

FUNCIONES HASHTAG

Hashtag permite la creación de funciones que retornan un valor, dato o función, donde este realizará una tarea específica. Así como creación de los que no retornan ningún valor. Se debe especificar el tipo, nombre y la declaración de parámetros de la función. La lista de parámetros son identificadores separado por comas (,).

Ejemplo

```
function int suma (int a, int b)  do
    return a + b;
end

function string saludo ( )  do
    return "Hola mundo";
end
```

LLAMADOS

A la hora de llamar una función, se escriben el nombre de la función y los parámetros entre paréntesis; pero en caso que no se indique los parámetros solo se incluye los paréntesis vacíos ( ).

Ejemplo:

```
Suma ( );

Suma ( 5, 5);
```

PRINTS

La palabra reservada "print" seguido de los paréntesis donde se especifica la salida.

Ejemplo

```
print ( "Se_cambio_el_valor_de_var2" );        {Print de un mensaje}

print ();                                      {print vacío}

print (a + b);                                 {print expresión}

print (Suma);                                  {print función Suma}
```

ESTRUCTURAS DE CONTROL DEL FLUJO

Son los que permiten tomar decisiones y realizar un proceso repetidas veces.

IF

Estructura que permite ejecutar un conjunto de sentencias en función del valor que tenga la expresión de comparación. Por ejemplo:

```
#IF ELSE
    if(condicion) #Expresion Booleana
    do
       #instrucciones que se ejecutan si la condicion es true
       else
       #instrucciones que se ejecutan si la condicion es false
    end

#IF ELSEIF ELSE
       if (condicion1) #Expresion Booleana
       do
           #instrucciones Hashtag1
           else if(condicion2)
           do
              #instrucciones Hashtag2
              else if (condicion3)
              do
                 #instruccion Hashtag3
                 else
                     #instruccion Hashtag4

               end
           end
       end

#IF COMPARACION
    if((diasemana >= 1) and (diasemana <=5))
    do
         trabajar = false;
       else do
           trabajar = true;
       end
    end
```

```
int usuario = 45;

if (usuario <= 18)
do
    print ("Usuario Menor");
    else
        if(usuario > 45)
        do
          print ("Usuario Mayor");

end
          end
    end
```

SWITCH

Esta es una alternativa del if else if else cuando se compara la misma expresión con distintos valores. En Hasthag se utilizan las palabras reservadas switch y case: donde cada case corresponde con único valor de expresión. Para finalizar se utiliza la palabra reservada break; indicando que se termina el switch.

```
3
4     switch(expresion)  do
5           case 1:
6             Instruccion;
7             break;
8            case 2:
9             Instruccion;
10            break;
11           case 3:
12            Instruccion;
13             break;
14              other:
15     end
16    switch(expresion)
17          case 1:
18
19            print("Primer Caso");
20
21            break;
22           case 2:
23            print("Segundo Caso");
24            break;
25           case 3:
26            print("Tercer Caso");
27             break;
                    other:
```

WHILE

La instrucción while permite crear bucles. Estos agrupan instrucciones las cuales se ejecutan continuamente hasta que una condición que se evalúa sea falsa.

Sintaxis While:

```
while (condicion)
do
    #sentencia que se ejecuta si la condicion es true
end
```

FOR

Es un bucle que se utiliza para ejecutar instrucciones controladas por un contador. Una herramienta muy útil que proporciona Hashtag. La sintaxis es la siguiente:

```
for (ExpresionIncial; Condicion; ExpresionEnCadaVuelta)
    do
       #Insrucciones;
    end
```

HASHTAG PROGRAMA

```
begin

    function int suma(int a, int b) do
        return a + b;
    end

    function string saludo () do
        return "Hola mundo";
    end

    mainbegin do    #mainbegin_do...end
            int var1 = -8;
            int var2 = 11;
            int var3 = 21;
            boolean b1 = false;
            int s1 = 3 + 5 + 1 - 5 * 3;
            string str = "Hola_mundo" ;
            double db = 0.32;

            #if_ception!!
            if (var == 0) do
                if (var1 == 9) do
                    result = var1+var2/var3-var2;
                end else do
                    result = var1 - var2 * var3;
                    if (result == 0) do
                        if (var == 0) do
                            if (var1 == 9) do
                                result = var1+var2/var3-var2;
                            end else do
                                result = var1 - var2 * var3;
                                if (result == 0) do
                                    print ( "Dio_0!" );
                                end
                            end
                        end
                        print ( "Dio_0!" );
                    end
                end
            end

```

```
42              print (suma(var1,var3));
43              print (str);
44
45              double mult = db * var2;
46              int a,b,c = 0;
47
48              contador = 0;
49
50              while (contador != 5) do
51
52                  for (int i = 0; i < 100; i++) do
53                      if (i % 7 == 0) do
54                          contador++;
55                      end
56                  end
57              end
58
59              switch (contador) do
60
61                  case 1: do
62                      print ( "Encontro_solo_1." );
63                      break ; #break_en_todo_case
64                  end #end_case
65
66                  case 2: do
67                      print ( "Encontro_2." );
68                      break ;
69                  end #end_case
70
71                  case 3: do
72                                  if (var == 0) do
73                                      if (var1 == 9) do
74                                          result = var1+var2/var3-var2;
75                                      end else do
76                                          result = var1 - var2 * var3;
77                                          if (result == 0) do
78                                              if (var == 0) do
79                                                  if (var1 == 9) do
80                                                      result = var1+var2/var3-var2;
81                                                  end else do
```

```
                                    result = var1 - var2 * var3;
                                    if (result == 0) do
                                        print ( "Dio_0!" );
                                    end
                                end
                            end
                            print ( "Dio_0!" );
                        end
                    end
                end
                print ( "Encontro_3." );
                break ;
            end

            case 4: do
                print ( "Encontro_4." );
                break ;
            end

            other : do
                while ( true ) do
                    if ( true ) do
                    end
                end
                break ;
            end

        end

        if (var1 == var2) do

            for (int i = 0; i < 10; i++) do
                var2 = var1 + i;
            end
            print ( "Se_cambio_el_valor_de_var2" );

        end

        #------------------comments----------------
        #real*entero=real
```

```
122             #simple_comment!
123
124             #-----------------comments---------------
125
126
127             int result;
128             if (var == 0) do
129                 if (var1 == 9) do
130                     result = var1+var2/var3-var2;
131                 end else do
132                     result = var1 - var2 * var3;
133                     if (result == 0) do
134                         print ( "Dio_0!" );
135                     end
136                 end
137             end
138 sn
139             b1 = true;
140             print ( "Hola_mundo" );
141         end
142     endmain
143 end
```