

Programación datos del libro, piensa en C:

Los datos que procesará una computadora ya sean simples o estructurados, se deben almacenar en casillas o celdas de memoria para utilizarlos posteriormente. A estas casillas o celdas de memoria se les asigna un nombre para reconocerlas: un identificador, el cual se forma por medio de letras, dígitos y el carácter de subrayado (_). Siempre hay que comenzar con una letra. El lenguaje de programación C distingue entre minúsculas y mayúsculas, por lo tanto, AUX y Aux son dos identificadores diferentes. La longitud más común de un identificador es de tres caracteres, y generalmente no excede los siete caracteres. En C, dependiendo del compilador que se utilice, es posible generar identificadores más grandes (con más caracteres). Cabe destacar que hay nombres que no se pueden utilizar por ser palabras reservadas del lenguaje C. Estos nombres prohibidos se presentan en la siguiente tabla.

Palabras reservadas del lenguaje C

Auto

Do

Goto

Signed

unsigned

Break

double

if

Sizeof

void

Case

else

int

Static

Volatile

char

Enum

Long

struct

While

const

extern register

switch

continue

Float

return

Typedef

Default

for

short

Union

Constantes:

Las constantes son datos que no cambian durante la ejecución del programa. Para nombrar las constantes utilizamos identificadores. Existen tipos de constantes de todos los tipos de datos, por lo tanto, puede haber constantes de tipo entero, real, carácter, cadena de caracteres, etc. Las constantes se deben definir antes de comenzar el programa principal, y éstas no cambiarán su valor durante la ejecución

del mismo. Existen dos formas básicas de definir las

constantes: `const int nu1 = 20; /* nu1 es una constante de tipo entero.`

`*/ const int nu2 = 15; /* nu2 es una constante de tipo entero.`

`*/ const float re1 = 2.18; /* re1 es una constante de tipo real.`

`*/ const char ca1 = 'f'; /* ca1 es una constante de tipo carácter. */`

Otra alternativa es la siguiente:

`#define nu1 20; /* nu1 es una constante de tipo entero.`

`*/ #define nu2 15; /* nu2 es una constante de tipo entero.`

`*/ #define re1 2.18; /* re1 es una constante de tipo real.`

`*/ #define ca1 = 'f'; /* ca1 es una constante de tipo carácter. */`

Otra forma de nombrar constantes es utilizando el método enumerador: enum. Los valores en este caso se asignan de manera predeterminada en incrementos unitarios, comenzando con el cero. enum entonces es útil cuando queremos definir constantes con valores predeterminados. A continuación, se presenta la forma como se declara un enum:

```
enum { va0, va1, va2, va3 }; /* define cuatro constantes enteras. */
```

Esta definición es similar a realizar lo siguiente:

```
const int va0 = 0;  
const int va1 = 1;  
const int va2 = 2;  
const int va3 = 3;
```

Tipos de dato:

Una vez que se declaran las variables, éstas reciben un valor a través de un bloque de asignación. La asignación es una operación destructiva. Esto significa que, si la variable tenía un valor, éste se destruye al asignar el nuevo valor. El formato de la asignación es el siguiente:

```
variable = expresión o valor;
```

Donde expresión puede representar el valor de una expresión aritmética, constante o variable. Observa que la instrucción finaliza con punto y coma; Analicemos a continuación el siguiente caso, donde las variables reciben un valor a través de un bloque de asignación.