Fundamentos de Algoritmos

Tema 1
Clase 4

Tipo de Datos:
Variables, Constantes, Operadores y Expresiones

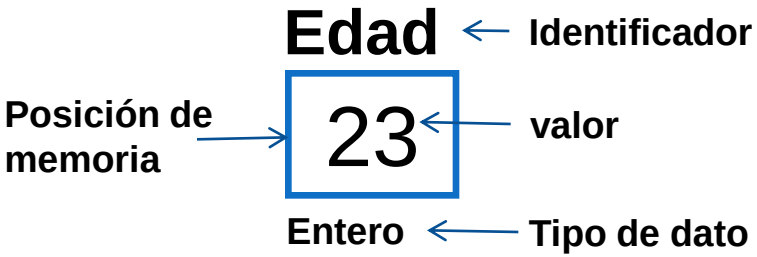
Variables

En programación, una variable es un espacio de memoria reservado para almacenar un valor que corresponde a un tipo de dato soportado por el lenguaje de programación. Por esta razón, una variable tiene un aspecto físico y un aspecto lógico.

El aspecto físico se concibe como el espacio de memoria donde la variable reside físicamente.

En cuanto al aspecto lógico, las variables tienen tres atributos:

- el nombre o identificador,
- el tipo (tipo de dato) y
- su valor.



Variables

El valor asociado puede cambiar durante la ejecución del programa. Su cambio no es arbitrario, sino producto de la ejecución de ciertas sentencias en el programa.

Por cada variable se reserva “una posición” en la memoria donde se aloja su valor corriente. Tal posición es solo accedida a través del nombre de la variable o identificador.

Variables

En lenguaje pseudoformal:

<tipo de dato> <nombre> = <valor>

Ejemplo:

entero Edad = 12

(valor de la inicialización, pero que puede cambiar)

otra forma:

<tipo de dato> <nombre> y luego <nombre> = <valor>

entero Edad
Edad = 12 (valor de la inicialización)

En lenguaje C se define como:

int Edad = 12;

o

int Edad;
Edad = 12;

Identificadores

Representan simbólicamente las localidades de los datos en la memoria. Sin los identificadores el programador tendría que conocer y usar direcciones de datos para manipularlos.

Cuando se tienen los identificadores de los datos, es el compilador (en el caso del lenguaje C) quien se encarga de seguir la pista de dónde se localizan físicamente.

Identificadores

Reglas para formar los identificadores:

Los diferentes lenguajes de programación utilizan diversas reglas para formar los identificadores.

En lenguaje C los únicos símbolos de nombre válidos son :

- las letras mayúsculas, minúsculas,
- los dígitos del 0 al 9 y
- el subrayado.

El primer carácter del identificador no puede ser un dígito y tampoco se recomienda utilizar el subrayado, para iniciar el nombre del identificador.

Identificadores

Reglas para formar los identificadores:

- Se trata de una única palabra (cada dato tiene un nombre distinto para poder referirnos a él unívocamente) que debe resultar significativa, sugiriendo lo que representa.

- Es necesario que el identificador no coincida con ninguna palabra reservada, esto es, nombres que se correspondan con palabras propias del lenguaje (instrucciones, estructuras de control, etc…).

- Ejemplos de identificadores válidos:
  Acumulador1, Nombre_Persona, Cedula_Identidad, Curso_2001,

- Ejemplo de identificadores inválidos:
  123, hola#, &cedula,

Constantes

Es un espacio de memoria reservado para almacenar un valor fijo.

Son valores de datos que no pueden cambiarse durante la ejecución de un programa.

Por ejemplo un programador necesita usar el valor de pi (3,1416) varias veces; es muy conveniente definir una constante al principio del programa y usarla.

Constantes

Tipos de constantes:

- **Literal**: es un valor expresado en forma explícita.
  Ejm. 3.1416

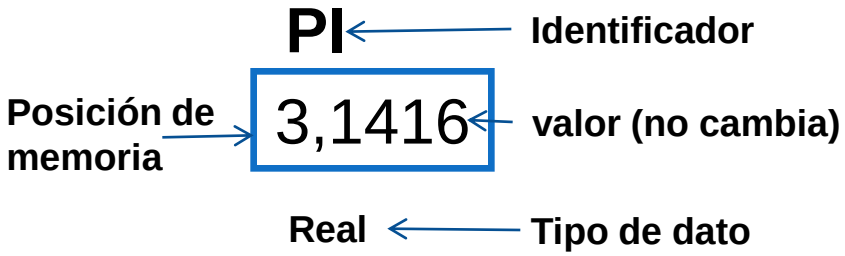
- **Simbólica**: viene expresado bajo un nombre que guarda su valor
  Ejm. Pi (Previamente se debe definir que Pi = 3.1416)

En lenguaje pseudoformal:

const <nombre> = <valor> (el tipo de la constante es definido por medio del valor asignado)

En lenguaje C se define como:

#define <NOMBRE> <valor>



Constantes

Uso de las constantes:

- Evita posibles errores de escritura cuando un dato se repite mucho a lo largo del programa.
- Clarifica la lectura del programa.
- Nos ahorra trabajo. Si queremos cambiar el valor del dato solamente tendremos que hacerlo en su definición.
- Los programas serán más fáciles de mantener y modificar.
- Ejemplos son: pi = 3.14159; iva = 12

Operadores

- El conjunto de operadores definido para un tipo de dato determina como se pueden manipular los datos.

- Pueden ser primitivos o definidos por el lenguaje de programación o definidos por el programador.

- Cada operador tiene:
  - un dominio (conjunto de valores posibles de entrada), y
  - el conjunto de posibles resultados que puede producir.

  para un conjunto dado de argumentos, la acción del operador, define el resultado que se produce.

Operadores

- Un operador maneja uno o varios datos (operandos) para realizar un determinado cálculo con ellos. Cuando maneja un solo dato se llama **unario**, si maneja dos datos, se llama **binario**.

- Dependiendo del tipo de dato de los operandos se podrán aplicar unos tipos de operadores u otros.

- Se conocen como los símbolos usados como función. Si tiene dos argumentos, la función generalmente es infija (se coloca en medio de los dos operandos) y si tiene solo un argumento es prefija .

Tipos de Operadores

- Relacionales
- Aritméticos
- Alfanuméricos
- Lógicos

Operadores Relacionales

Realizan operaciones de comparación entre distintos tipos de datos. El resultado es un valor lógico (verdadero o falso). Generalmente sirven para expresar condiciones en los algoritmos.

Operador	Significado
=	Igual que
<	Menor que
≤	Menor o igual que
>	Mayor que
≥	Mayor o igual que
≠	Distinto que

Operadores Relacionales

Comparaciones

El formato general para las comparaciones es :

Expresión1 <operador de relación> Expresión2

- Las operaciones de relación se pueden aplicar a cualquiera de los 5 tipos de datos vistos (enteros, real, lógico, carácter y cadenas).

Operadores Relacionales

Comparaciones con tipo de dato carácter

- Como los caracteres están situados en un orden creciente en el código ASCII, este orden se utiliza para compararlos y decidir si uno de ellos es menor o mayor que otro.

  ‘A’ < ‘a’ ≡ v

Operadores Relacionales

Comparaciones con tipo de dato lógico

- Cuando se utilizan los operadores de relación con valores lógicos la constante falso es menor que la constante verdadero

falso < verdad

verdad > falso

Operadores Aritméticos

Realizan operaciones entre datos numéricos. Su resultado es un valor numérico.

Operador	Significado
-	Menos unario
*	Multiplicación
/	División real
**	Exponenciación
+	Adición (Suma)
–	Resta
mod	Modulo (residuo) de la división entera
div	Cociente de la división entera

Operadores lógicos

Realizan operaciones entre datos lógicos o booleanos. El resultado es un valor lógico. El orden de precedencia es $\neg$, $\wedge$, $\vee$

Operador	Significado
no ($\neg$)	Negación (unario)
y ($\wedge$)	Conjunción (binario)
o ($\vee$)	Disyunción(binario)

Están basados en el Álgebra de Boole, donde cada operador, pose una especificación llamada tabla de verdad.

Resumen de los Operadores

OPERADORES	INTERVIENEN	OPERADORES	RESULTADO
ARITMETICOS	Datos Numéricos	Aritméticos +, - , *, /, div, mod, **	Dato Numérico
DE COMPARACION	Datos del mismo tipo	Relacionales >, <, >=, <=, =, ≠	Dato Lógico
LOGICOS	Datos lógicos	Lógicos ¬, ∧, ∨	Dato lógico

Precedencia de los Operadores Aritméticos y Relacionales

Niveles de prioridad o precedencia para evaluar expresiones aritméticas o relacionales.

Alto ↑ ↓ Bajo	**Nivel**	**Símbolo**
	1°	**
	2°	*, /, MOD, DIV, - (unario)
	3°	+, -
	4°	<, >, ≤ :, ≥ , =, ≠

Precedencia de los Operadores lógicos

Operadores lógicos

Alto ↑ ↓ Bajo	**Nivel**	**Símbolo**
	1°	¬
	2°	∧
	3°	∨

Precedencia de los Operadores Aritméticos Relacionales y lógicos

En resumen

Alto ↑ ↓ Bajo	**Nivel**	**Símbolo**
	1°	**, ¬
	2°	*, /, MOD, DIV, - (unario), ^
	3°	+, -, ∨
	4°	<, >, ≤ , ≥ , =, ≠

Expresiones

Una expresión es una combinación de una o varias operaciones para realizar un determinado cálculo.

Los cálculos combinando operandos con operadores son expresiones.

Combinan constantes, variables, operadores, paréntesis y nombre de funciones especiales.

Cada expresión tiene un valor que se determina tomando los valores de las variables y constantes implicadas y ejecutando las operaciones indicadas.

Expresiones Aritméticas

Utilizan datos y operadores numéricos y el resultado es un valor numérico.

Son análogas a las fórmulas matemáticas. Las variables y constantes son numéricas (entero o real) y las operaciones son las aritméticas clásicas.

Ej.

K * (3 + L) / (C - P)

15 MOD 5

Los paréntesis se utilizan para agrupar términos finitos y asegurar que las operaciones se ejecutan en el orden correcto.

Expresiones lógicas o Booleanas

Construidas con datos y operadores booleanos, el resultado es un valor booleano.

Combinan constantes y expresiones lógicas por medio de operadores lógicos ($\neg$, $\wedge$, $\vee$) y los operadores relacionales (<, >, $\geq$, ..). Ej.

$$(F > E) \wedge (R \leq E * 4)$$

Parentización de Expresiones

Las expresiones que tienen dos o mas operadores requieren reglas matemáticas que permitan determinar el orden de las operaciones.

1. Las operaciones encerradas entre paréntesis se evalúan primero. Si existe anidamiento de paréntesis se evalúan primero las expresiones mas internas.

2. Las operaciones aritméticas dentro de una expresión suelen seguir el siguiente orden:

   **

   *, /, DIV, MOD

   +, -

3. En caso de coincidir varios operadores de igual prioridad en una expresión se evalúan de izquierda a derecha.

Ejercicios

Evaluar las siguientes expresiones:

a.-(5/2)+(20*6)+7 mod 5
b.-((4*6)/2)-(15/2)
c.-((5*15)/2)/(4-2)
d.-(8=16)$\vee$(7$\neq$ 4)$\wedge$(4<1)
e.-(((4*3)<6)$\vee$(3>(5-2)))$\wedge$((3+2)<12)

- Ejemplo:

A= 3 ; B= 5 ; C= 4 ; D=2 ; E=1

8*A + (-B/5 + C) – 4 mod D > (A+5*(B*(1/-E)))

Precedencia de los Operadores en lenguaje C

Tipo de Operador	Orden de prioridad
Unarios	!, (Signos: -, +), ++, --
Multiplicativos	*, /, %
Aditivos	+, -
Relacionales	>, <, >=, <=, ==, !=
^	&&
v	||