



TECNOLÓGICO
NACIONAL DE MÉXICO



INSTITUTO TECNOLÓGICO DE NUEVO LAREDO

“Con la Ciencia por la Humanidad”

Introducción a la Ingeniería en Sistemas Computacionales y al Diseño Orientado a Objetos

Curso propedéutico

Instructor:

Bruno López Takeyas

bruno.lt@nlaredo.tecnm.mx

4.- Datos y tipos de datos

- 4.1 Conceptos de datos y tipos de datos
 - 4.2 Datos numéricos
 - 4.3 Datos lógicos (booleanos)
 - 4.4 Datos alfanuméricos
 - 4.5 Variables
 - 4.6 Constantes
-

La función principal de un sistema de procesamiento de información es recibir datos, procesarlos y convertirlos en información valiosa para la solución de problemas a través de computadoras. Sin embargo las computadoras manejan los datos, según las necesidades, según los valores que almacenan y de acuerdo a un formato requerido, el cual determina el conjunto de operaciones que pueden realizarse con ellos.

Conceptos de datos y tipos de datos

Un dato es una representación simbólica de una característica o atributo de una cosa, objeto o entidad y pueden expresarse en diferentes formatos; mientras que un tipo de dato es la clasificación de ciertos datos que cumplen con ciertas características o formato de acuerdo a los valores que almacenan. Esto significa que el dato es el valor de una característica de una cosa, mientras que el tipo de dato es el formato con el que se presenta dicho valor.

Existen dos clases de tipos de datos: simples (sin estructura) y compuestos (estructurados). Los tipos de datos simples solamente almacenan un valor a la vez, mientras que los tipos de datos compuestos o estructurados pueden contener muchos valores al mismo tiempo. En esta sección se hace referencia a los tipos de datos simples.

Tipos de datos simples

Los tipos de datos simples son los siguientes:

- Numéricos (ENTERO, REAL)
- Lógicos (BOOLEANO)
- Alfanuméricos (CHAR, CADENA o STRING)

Datos numéricos

Este tipo de datos sólo puede almacenar números y pueden presentarse de dos formas:

- Enteros**
- Reales**

Los enteros son números completos, es decir, no tienen componentes fraccionarios o decimales y pueden ser negativos, positivos o cero.

Es importante mencionar que una computadora tiene ciertas reglas para manejar sus tipos de datos, por ejemplo, un dato numérico entero no puede contener punto decimal (aunque su parte fraccionaria sea cero), símbolos especiales (como el símbolo para indicar un tipo de moneda), ni la coma separadora de las cifras

Ejemplos de datos numéricos enteros

| | | |
|------|-----|-----|
| 5 | 6 | -15 |
| 4 | 20 | 0 |
| 1340 | -23 | -1 |

Ejemplos válidos de datos numéricos enteros.

| | | |
|-------|--------|--------------------|
| 5.2 | \$6 | -5×10^2 |
| 4.0 | 12,320 | 0.0 |
| 1,340 | -2,300 | 8×10^{-3} |

*Ejemplos **NO** válidos de datos numéricos enteros.*

Datos numéricos enteros con notación exponencial

También se puede usar la notación científica o exponencial para representar datos numéricos enteros muy grandes o muy pequeños, donde se tiene un coeficiente que se multiplica por una potencia de 10. Para representar un número entero con notación científica, se sustituye la potencia de 10 por la letra E (mayúscula). Por ejemplo, el valor 4×10^3 (cuyo valor representa el número 4000), en una computadora se utiliza el formato 4E3; o bien el valor 8×10^{-3} se representa como 8E-3.

Datos numéricos reales

Los números reales siempre tienen un punto decimal y pueden ser negativos, positivos o cero.

| | | |
|-----------|---------|--------|
| 0.08 | 3739.41 | 3.7452 |
| -52.321 | -8.12 | 0.0 |
| 3.1415927 | -5.1 | 63.9 |

Ejemplos válidos de datos numéricos reales.

| | | |
|-------------|------------|---------|
| 0.089,073 | \$3,739.00 | £3.7452 |
| -52,476.321 | \$8.12 | ¥120.50 |
| 3.141,592,7 | -µ5.1 | €63.9 |

*Ejemplos **NO** válidos de datos numéricos reales.*

Datos numéricos reales con notación exponencial

También se pueden representar datos numéricos reales utilizando la notación científica o exponencial, por ejemplo el valor 5.2×10^5 , en una computadora se representa como 5.2E5; o bien, el valor 6.435×10^{-3} , se representa como 6.435E-3.

Es importante mencionar que un dato numérico real acepta que se le suministren datos numéricos enteros, pero al recibirlo y almacenarlo, internamente lo representa con el punto decimal y le asigna un cero a su parte fraccionaria.

Datos lógicos (booleanos)

El tipo de dato lógico (también conocido como booleano) es aquel que solo puede tomar uno de los siguientes valores: **Verdadero** o **Falso**. Algunos autores manejan estos valores como equivalentes: **si**, **verdadero**, **cierto**, **true**, o le asignan el valor numérico de **1 (uno)** y **no**, **falso**, **false** o le asignan el valor numérico de **0 (cero)**.

Cuando se asigna un valor diferente a verdadero o falso a un dato de tipo booleano, entonces se presenta un error.

Datos alfanuméricos

Esta clasificación de datos pueden aceptar números, letras o símbolos especiales y se clasifican en dos:

- Caracter (CHAR)
- Cadenas (CADENA o STRING).

Datos tipo caracter

Este tipo de datos maneja los símbolos de la computadora. Un dato tipo caracter contiene un solo símbolo o caracter. Los caracteres reconocidos por las computadoras son:

Alfabéticos: 'A', 'B', 'C', 'Z', 'a', 'b', 'c', , 'z'

Numéricos: '0', '1', '2', '3', , '9'

Caracteres especiales: '+', '-', '*', '\$', '#', '@', '&', '€', '∞', 'Ω'

Los datos de tipo caracter se representan entre apóstrofes (') y almacenan sólo un símbolo.

Ejemplos de datos de tipo caracter

| | | |
|-----|-----|-----|
| 'A' | '5' | '£' |
| 'a' | ' ' | '¥' |
| 'á' | '9' | '¿' |

Ejemplos válidos de datos de tipo caracter.

| | | |
|-------|---------|-------------|
| 'AB' | '50' | '£100.00' |
| 'ab' | '100' | '¥1,234.56' |
| 'ábc' | "9,000" | "¿Qué?" |

*Ejemplos **NO** válidos de datos de tipo caracter.*

Notas con datos de tipo caracter

Es importante destacar que un dato de tipo caracter no puede aplicarse en operaciones aritméticas (aunque su valor “parezca” un número entero); es decir, el dato ‘5’ no representa la cantidad numérica entera del 5, sino la representación simbólica que corresponde a un caracter (nótese que el valor se representa entre apóstrofes).

Datos de tipo cadena o string

Una cadena o *string* es una sucesión de caracteres que se encuentran delimitados por comillas ("). La Fig. 5.7. muestra varios ejemplos válidos de cadenas o *strings*.

| | | |
|-----------------|------------------|-----------|
| "Tec Laredo" | "(867) 714-0022" | "£100.00" |
| "Pedro Infante" | "\$1,200.00" | "" |
| "Reforma 2007" | "9,000" | "¿Qué?" |

Ejemplos válidos de datos de tipo cadena o string.

Notas con datos de tipo cadena

Obsérvese que esta sucesión de datos puede contener representaciones simbólicas de letras, símbolos o números. Los datos representados entre comillas, no pueden utilizarse para operaciones aritméticas (aunque “parezcan” ser datos numéricos). P. ejem. Los datos “1234”, “5678.90”, “\$ 100.00”, “1,234. 56” no representan datos numéricos sino cadenas (nótese que se representan entre comillas).

Para indicar una cadena vacía, basta con poner las comillas juntas (“”); pero si se deja un espacio entre ellas (“ ”), entonces la cadena contiene un espacio en blanco y no está vacía, ya que el espacio en blanco se considera un carácter válido.

Variables

Una variable es una localización o celda en la memoria principal (RAM) que almacena un valor que puede cambiar en el transcurso de la ejecución del programa. Cuando un programa necesita almacenar un dato, entonces requiere de una variable. Toda variable tiene un nombre, un tipo de dato y un valor. Antes de poder utilizar una variable es necesario declararla especificando su nombre y su tipo de dato.

Declaración de variables

Declaración de una variable:

tipo nombre

P. ejem.

ENTERO EdadEstudiante

REAL PorcentajeDescuento

CHAR Grupo

STRING NombreEmpleado

Formato de declaración de variables.

Variables

Si se intenta asignar un valor de un tipo a una variable de otro tipo se producirá un error de tipo. Los nombres de las variables se forman con caracteres alfanuméricos, de los cuales el primero debe ser una letra y puede estar seguida de letras, números o guiones bajos. En la actualidad, se recomienda definir nombres de variables que sean representativos de su contenido y para ello se sugiere utilizar la notación de Pascal para nombrar variables, la cual consiste en declarar variables cuyas iniciales sean mayúsculas.

Variables

| | | |
|---------------|----------------|---------------|
| SubTotal | ClaveEmpleado | Domicilio |
| PorcentajeIVA | SueldoEmpleado | PlacasAuto |
| NombreAlumno | EdadPersona | Resistencia23 |

Ejemplos válidos de nombres de variables.

| | | |
|-------|--------------|-------|
| X-Y | Precio\$ | a-b-c |
| 2X | €Euros | A |
| W 3.4 | Edad Persona | \$q23 |

*Ejemplos **NO** válidos de nombres de variables.*

Constantes

Son constantes todos aquellos valores fijos que figuran en los algoritmos; es decir, no cambian como producto de procesamiento. Existen numerosas constantes en el mundo de las matemáticas, por ejemplo, π (3.1416), e (2.71828), etc.

Al igual que las variables, la definición de los nombres de las constantes también debe cumplir las reglas de nomenclatura y el formato de declaración. Cuando se declara una constante, se coloca la palabra **CONST**, seguido del tipo de dato, el nombre de la constante y debe asignársele su valor.

Declaración de constantes

Declaración de una constante:

```
CONST tipo nombre
```

P. ejem.

```
CONST REAL Pi = 3.1415927
```

```
CONST ENTERO PorcentajeIVA = 16
```

Formato de declaración de constantes.

Prácticas

■ Descargue del sitio web:

<https://nlaredo.tecnm.mx/takeyas/LibroISC>

■ *Práctica 4.1.- Tipos de datos*



Tarea

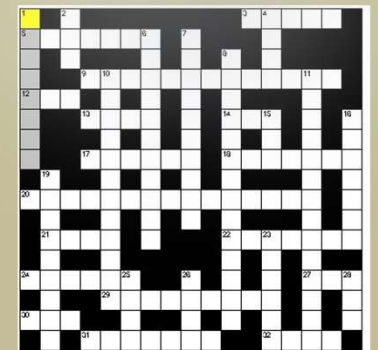
Resuelva en el sitio web

<https://nlaredo.tecnm.mx/takeyas/LibroISC>

Cuestionario 4.1



Crucigrama 4.1



Fuentes de información:

- López Takeyas, Bruno. (2019) “Introducción a la Ingeniería en Sistemas Computacionales y al diseño orientado a objetos”. Editorial Pearson.
- <https://nlaredo.tecnm.mx/takeyas/LibroISC/>

