



Fundamentos de Programación

Plan de Acompañamiento y Asesoría Estudiantil

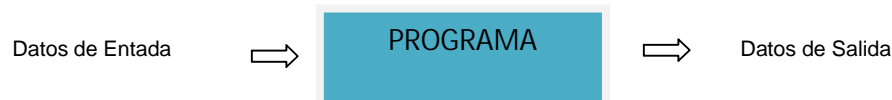
FUNDAMENTOS DE PROGRAMACIÓN

UNIDAD 2: VARIABLES Y TIPOS DE DATOS

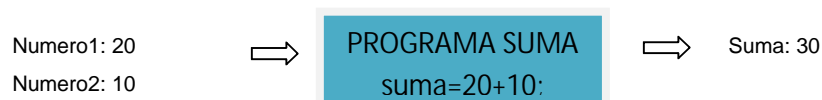
OBJETIVOS:

- Introducir los conceptos básicos sobre variables, y tipos de datos.
- Aprender a utilizar tipos de datos y variables dentro de la resolución de un problema.

El primer objetivo de toda computadora es el manejo de la información o datos. Estos datos serán los valores de los actores que nos presenta el problema específico a solucionar, Los programas y los Algoritmos correspondientes operan sobre estos tipos de datos.



Es importante en el proceso de análisis del problema para su posterior resolución (Diseño de Algoritmo) identificar los datos que operará para obtener resultados. De esta manera basta con identificar que se operará, y con los que operará los procesos dentro del programa para así mismo identificar cuáles serán los resultados a partir de los datos de entrada proporcionados

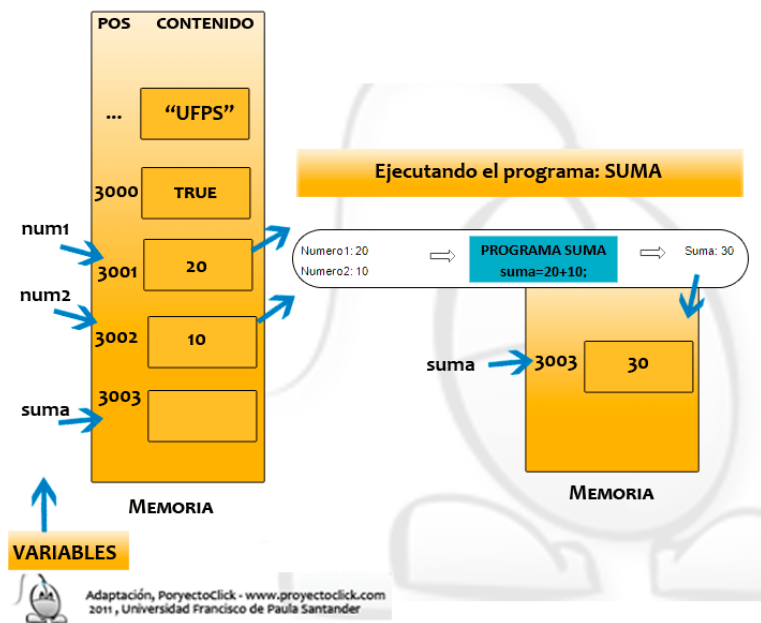


Qué es un dato? Es la representación formal de hechos, conceptos, o instrucciones adecuada para su comunicación, interpretación y procesamiento. Los datos es la representación simbólica (numero, letra, u otros) mínima de una característica



Qué es una variable? Una variable no es más que un nombre simbólico que identifica una dirección de memoria, para almacenar un valor de un tipo determinado, a lo largo de la ejecución del programa la variable es accesible, tantas veces como se necesite.

La utilización de variables es necesaria para la operación de programas:



- El programa suma crea tres posiciones de memoria: 3001, 3002, y 3003.
- Dentro de la 3001 y 3002 almacena los datos de entrada (los que ingresa el usuario.) (20 y 10)
- En la posición 3003 efectúa sus procesos. (Almacenar el resultado de la posición 3001, y 3002). Guardando en 3003=30.

Pero el programa suma para identificar esas posiciones de memoria donde almacena datos, utiliza variables que simplemente identificarán esos espacios donde tiene almacenados los datos.

Tipos de datos.

Teniendo entonces la estrecha relación entre variables y datos. Se definen tipos de datos, para identificar variables que solo almacenarán datos según la clasificación de los datos a tratar. Si solo se tratarán números, letras, o según sea el caso.

En el computador los datos se almacenan en binario. (0, o 1). Y las distintas clasificaciones de los tipos de datos nos permiten definir el tipo y tamaño máximo que incluye.



Todo lenguaje de programación define tipos de datos y encontramos los siguientes tipos de datos primitivos para Java(lenguaje de programación de alto nivel a estudiar):

TIPO DE DATO	TAMAÑO(BITS)	RANGO
byte	8	-128 a 127
short	16	-32768 a 32767
int	32	-2147483648 a 2147483647
long	64	-9223372036854775808 a 9223372036854775807
float	32	+/- 3.4E+38F (6-7 Dígitos importantes) 8 bits exponente / 23 dígitos
double	64	+/- 1.8E+308 (15 Dígitos importantes) 11 bits exponente / 53 dígitos
char	16	Conjunto caracteres Unicode
boolean	1	true, false (verdadero o falso)
string	--	--

	<p>ENTEROS. byte, short, int, long Utilizados para declarar tipos de datos enteros específicos. Ejemplo: edades, horas, entre otros. (Sus tamaños dependen de que tantos números quieran almacenarse). Lo mas utilizados: byte e int</p>
	<p>DECIMALES float, double Utilizados para declarar tipos de datos con valores decimales. Ejemplo: salarios, valores, medidas, grados. Dependen de la representación coma flotante definida para los PCs Lo mas utilizados: float (cálculos normales) doubles(cálculos grandes)</p>
	<p>CARACTERES char Utilizados para almacenar un carácter (letra o símbolo) Ejemplo: "F", "M", "a", "@" Depende de la codificación UNICODE (representación caracteres) (SOLO DE A UNO)</p>



Fundamentos de Programación

Plan de Acompañamiento y Asesoría Estudiantil

	<p>BOOLEAN</p> <p>boolean</p> <p>Utilizados expresiones con valor boolean.. Verdadero o Falso</p> <p>Ejemplo: preguntas, consultas, fuma? ,Bebe?.</p> <p>(Sus únicos valores son: true(1) o false(0))</p>
	<p>TEXTOS</p> <p>String</p> <p>No se define como tipo de dato, aunque se utiliza como el mismo</p> <p>Ejemplo: nombre, mail, dirección</p> <p>Es un conjunto de tipos de datos Char (que solo almacena una cadena)</p> <p>Se utilizan colocando el valor en comillas: "ProyectoClick", "UFPS"</p>

Declaración de variables. Para utilizar las variables en un programa hay que declararla. Primero se declaran las variables y luego se inicia las lecturas o procesos de la misma.

- Se llama identificador, al nombre de la variable y la forma en que nos vamos a referir a ella en el programa.
- Al declarar una variable hay que definir su tipo, y solo almacenará datos del tipo especificado
- En una misma declaración se puede declarar varias variables, siempre que estas sean del mismo tipo. En este caso los identificadores se separarían por comas.

<Tipo de dato> identificador;

<Tipo de dato> lista de identificadores;

Ejemplo:

- Declarando una variable para guardar la edad. Tipo: int
int edad;
- Declarando una variable para guardar el sueldo. Tipo: float
float sueldo;



Fundamentos de Programación

Plan de Acompañamiento y Asesoría Estudiantil

- Múltiples variables: num1, num2, num3. Tipo: int

int num1,num2,num3;

Los **identificadores** tendrán las siguientes condiciones:

- Deberá comenzar por una letra (a...z / A...Z) pero no por un número.
- Después del primer carácter puede ir cualquier letra o número (a...z/A...z/0...9) signo \$ o _
- Las mayúsculas y las minúsculas se consideran diferentes
- El identificador debe ser descriptivo debe hacer referencia al significado de lo que se refieren.
- Suelen ser en minúsculas
- Cuando está formado por varias palabras, la primera va en minúscula y el resto de palabras se inicia con una letra mayúscula

Ejemplos Validos:

edad, usd\$, num1, numero_2, sueldoEmpleado

Ejemplos No Validos:

345numero, mi variable, cañada, Nombre.Largo, em@il

Inicialización de Variables.

En una declaración las variables pueden tomar un valor asignado por uno como por primera vez. Esto es útil cuando esa variable se va a usar en una expresión a evaluar en el proceso del programa. De la misma forma que se inicializa una variable, se le asigna un valor a ella.

<Tipo de dato> identificador = <valor inicial>;

int num1 = 0;



Constantes.

Una constante hace referencia a un valor que no puede modificarse. Su valor siempre será el mismo.

final <Tipo de dato> identificador = <valor inicial>;

Se define la constante igual que cuando se declara una variable y se inicializa, solo que se utiliza la palabra reservada final antes de la declaración lo que impide su modificación en el programa.

Los identificadores asociados a las constantes se suelen colocar en mayúsculas,

final double PI = 3.141592;

Si el identificador está conformado por varias palabras estas se separarán con el signo “_”

Referencias:

- [1] MIT OPEN COURSEWARE, Massachusetts Institute Of Technology – Introducción a la informática y a la resolución de problemas de Ingeniería. 2002 [Online]
<http://mit.ocw.universia.net/1.00/s02/class-sessions/>
- [2] CEBALLOS Fco. JAVIER (2006). *Java2: Curso de Programación*. 3ª Edición. Mexico: Alfaomega Ra-Ma
- [3] JOYANES, L. (2008). *Fundamentos de la programación. Algoritmos y Estructura de Datos*, 3ª Edición. Madrid: McGraw-Hill.
- [4] PROYECTO CUPÍ2 – Universidad De Los Andes [Online] <http://cupi2.uniandes.edu.co>
- [5] BERZAL GALIANO, FERNANDO - "Apuntes de programación orientada a objetos en Java: Fundamentos de programación y principios de diseño". Universidad de Granada, Granada, España [Online] <http://elvex.ugr.es/decsai/java/>