

Fundamentos de Programación

Unidad 2.

Introducción a la programación



Unidad 2. Introducción a la programación

2.1 Conceptos básicos.

2.2 Características del lenguaje de programación.

2.3 Estructura básica de un programa.

2.4 Elementos del lenguaje:

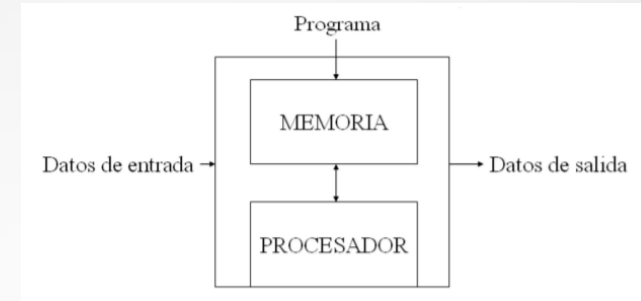
- tipos de datos,
- literales,
- constantes,
- variables,
- identificadores,
- palabras reservadas,
- parámetros,
- operadores
- salida de datos y entrada.

2.5 Traducción de un programa: compilación, enlace, ejecución y errores.

2.1 Conceptos básicos

Computadora:

Sistema en el que se introducen datos para ser procesados, realizando operaciones sobre estos o transformándolos y devolver el resultado de dicho procesamiento.



Problema:

Un problema es una proposición encaminada a averiguar el modo de obtener un resultado, cuando se conocen ciertos datos de partida.



2.1 Conceptos básicos



Tipos de problemas:

Irresolubles:

No pueden ser resueltos con los recursos disponibles.

Determinados:

Tienen una única solución.

Indeterminados:

Tienen muchas soluciones posibles.

2.1 Conceptos básicos

Algoritmo:

Descripción precisa de los pasos que se deben seguir para solucionar un problema planteado.



Lenguajes algorítmicos:

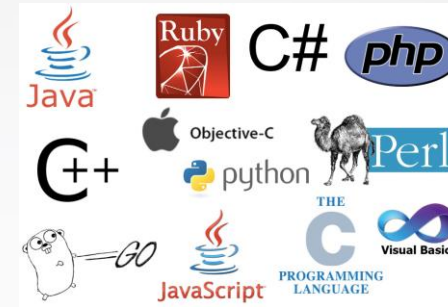
Sirven para describir un algoritmo en términos comprensibles para las personas.

```
1 Proceso ORDENAR_NUMEROS
2   Escribir "INGRESE LOS NUMEROS";
3   Leer N1, N2, N3;
4   Si N1 > N2 Entonces
5     Si N2 > N3 Entonces
6       Escribir "LOS NUMEROS ORDENADOS ES=", N3, ", ", N2, ", ", N1;
7     Sino
8       Si N1 > N3 Entonces
9         Escribir "LOS NUMEROS ORDENADOS ES=", N2, ", ", N3, ", ", N1;
10      Sino
11        Escribir "LOS NUMEROS ORDENADOS ES=", N2, ", ", N1, ", ", N3;
12      FinSi
13    FinSi
14  Sino
15    Si N1 > N3 Entonces
16      Escribir "LOS NUMEROS ORDENADOS ES=", N3, ", ", N1, ", ", N2;
17    Sino
18      Si N2 > N3 Entonces
19        Escribir "LOS NUMEROS ORDENADOS ES=", N1, ", ", N3, ", ", N2;
20      Sino
21        Escribir "LOS NUMEROS ORDENADOS ES=", N1, ", ", N2, ", ", N3;
22      FinSi
23    FinSi
24  FinSi
25 FinProceso
```

2.1 Conceptos básicos

Programa:

Conjunto de instrucciones precisas comprensibles para las computadoras, en un lenguaje de programación.



Programación:

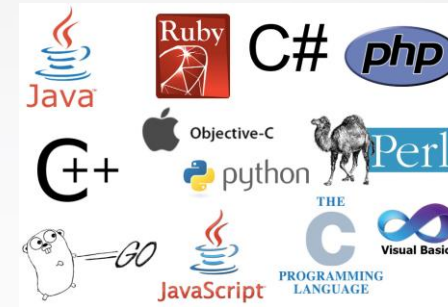
Todo el proceso que se lleva a cabo para construir un programa.

```
Análisis del problema
Especificación del problema
Diseño del algoritmo
Prueba de escritorio
Desarrollo del programa
Compilación/Interpretación
Ejecución
Verificación
```

2.1 Conceptos básicos

Lenguaje de programación:

Lenguaje diseñado para representar algoritmos para la computadora, con una sintaxis y semántica más sencillas que un lenguaje natural.



Sintaxis:

Reglas para la formación de todas las estructuras de un programa.

Semántica:

Significado de cada tipo de estructura formada en un programa.

2.2 Características del lenguaje de programación Java

Es SIMPLE: ...
Es ORIENTADO A OBJETOS: ...
Es DISTRIBUIDO: ...
Es ROBUSTO: ...
Es de ARQUITECTURA NEUTRAL: ...
Es SEGURO: ...
Es PORTABLE: ...
Es INTERPRETADO:



Java es un lenguaje orientado a objetos, aunque no de los denominados puros; en Java todos los tipos, a excepción de los tipos fundamentales de variables (int, char, long...) son clases.

2.3 Estructura básica de un programa en Java



```
package estructuradeunprograma;
```

```
public class Clase
```

```
{
```

```
    public static void main(String[] args)  
    {
```

```
        //instrucciones ejecutables
```

```
    }
```

```
}
```

Encabezado de la clase

Delimitadores de un bloque

Cuerpo de la clase

2.3 Estructura básica de un programa en Java



```
package estructuradeunprograma;
```

```
public class Clase  
{
```

```
    public static void main(String[] args)
```

```
    {
```

```
        //instrucciones ejecutables
```

```
    }
```

```
}
```

Encabezado del método

Cuerpo del método

Delimitadores de un bloque

2.4 Elementos del lenguaje Java



Tipos de datos

Es la propiedad de un valor, que determina que valores puede tomar, que operaciones se le pueden aplicar y como es representado.

TIPO	LONGITUD	NATURALEZA	RANGO DE VALORES
boolean	sólo usa 1 bit	lógico	true / false
char	16 bits	carácter	Unicode
byte	8 bits	entero	- 128 ... + 127
short	16 bits	entero	- 32.768 ... + 32.767
int	32 bits	entero	- 2.147.483.648 ... + 2.147.483.647
long	64 bits	entero	- 9.223.372.036.854.775.808... + 9.223.372.036.854.775.807
float	32 bits	real	coma flotante de simple precisión Norma IEEE 754
double	64 bits	real	coma flotante de doble precisión Norma IEEE 754



2.4 Elementos del lenguaje Java



Literales

Los valores literales son valores que se pueden asignar a una variable o constante, estos pueden ser números, caracteres o cadenas de caracteres.

números	caracteres	Cadenas de caracteres
45	'a'	"Ingrese un valor"
23.11	'm'	"Hola mundo"
0.153	'f'	"Nombre: "
0	'\n'	"Total a pagar: "

2.4 Elementos del lenguaje Java



Constantes

Es una variable cuyo valor no cambia durante la ejecución del programa. En Java se definen mediante el modificador final.

final tipo_constante identificador = valor;

```
final int DIAS_SEMANA = 7;
```

```
final int MESES = 12;
```

```
final char FEMENINO = 'F';
```

```
final double CALIFICACION_MINIMA = 5.5;
```

```
final boolean ACEPTADO = true;
```



2.4 Elementos del lenguaje Java



Variables

Es el nombre dado a una ubicación de memoria. El valor almacenado en una variable se puede cambiar durante la ejecución del programa. En Java, todas las variables deben declararse antes de que puedan ser utilizadas.

tipo_variable identificador;

```
int numero = 2;  
String cadena = "Hola";  
double altura = 2.4;  
boolean flag = true;
```



2.4 Elementos del lenguaje Java



Identificadores

Son los nombres que el programador asigna a variables, constantes, clases, métodos, paquetes, etc. de un programa. Están formados por letras y dígitos. Deben observar las siguientes reglas:

- *Los caracteres permitidos son: todos los caracteres alfanuméricos, además de (,), "\$" (signo de dólar) y '_' (guión bajo).*
- *No deben comenzar con dígitos.*
- *Distinguen entre mayúsculas y minúsculas.*
- *No hay límite en la longitud del identificador, pero se recomienda de 4 a 15 caracteres.*
- *Las palabras reservadas no se pueden usar como un identificador.*

2.4 Elementos del lenguaje Java



Palabras reservadas

Son algunas palabras definidas por el lenguaje para representar o realizar ciertas tareas previstas. No pueden ser utilizadas para otro fin, no pueden ser utilizadas como nombres de variables, clases o métodos, etc.

Abstract	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while



2.4 Elementos del lenguaje Java



Parámetros

Los parámetros se utilizan en los métodos, son variables que almacenan los valores que el método recibe por parte del código que le llama. En la declaración del método se escriben después del nombre, entre paréntesis, indicándose el tipo de cada uno y su nombre.

```
public void Alumno(String numCtrl, String nombre, int promedio){  
  
}  
  
public void otroMétodo() {  
    new Alumno("55050618", "Mickey", 83);  
}
```



2.4 Elementos del lenguaje Java



Operadores

Operadores	Asociatividad	Descripción
(<expresión>), (<list>), [<expresión>], <type>	Izquierda a derecha	Paréntesis, parámetros, índices, acceso a miembros
++ , -- , + , - , ! , ~ , new <nombredeclase> , (<type>)	Derecha a izquierda	Incremento unitario, decremento unitario, más, menos, inversión lógica, inversión de bit, instanciación de un objeto, tipo cast
* , / , %	Izquierda a derecha	Multiplicación, división, residuo
+ , - , +	Izquierda a derecha	Suma, resta, concatenación de cadenas
<< , >> , >>> ,	Izquierda a derecha	Desplazamiento de bits
< , <= , > , >= , <objeto> instanceof <class>	Izquierda a derecha	Comparaciones de intervalos, se ajusta a
== , !=	Izquierda a derecha	Igual, no es igual (diferente)
&	Izquierda a derecha	Ambos (bits)
^	Izquierda a derecha	Uno de los dos pero no ambos (bits)
 	Izquierda a derecha	Uno de los dos o ambos (bits)
&&	Izquierda a derecha	ambos
 	Izquierda a derecha	Uno de los dos o ambos
? :	Derecha a izquierda	Operador condicional
= , += , -= , *= , /= , %= , <<= , >>= , >>>= , &= , ^= , =	Derecha a izquierda	Asignación

2.4 Elementos del lenguaje Java



Entrada de datos:

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class EntradaJava
{
    public static void main(String[] args) throws IOException
    {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Por favor ingrese su nombre");
        String nombre = br.readLine();
        System.out.println("Bienvenido " + nombre + ". Por favor ingrese su edad");
        String entrada = br.readLine();
        int edad = Integer.parseInt(entrada);
        System.out.println("Gracias " + nombre + " en 10 años usted tendrá " + (edad + 10) + "
años.");
    }
}
```



2.4 Elementos del lenguaje Java



Entrada de datos:

```
import java.util.Scanner;

public class EntradaJava
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Por favor ingrese su nombre");
        String nombre = sc.nextLine();
        System.out.println("Bienvenido " + nombre + ". Por favor ingrese su edad");
        int edad = sc.nextInt();
        System.out.println("Gracias " + nombre + " en 10 años usted tendrá " + (edad + 10) + " años.");
    }
}
```



2.4 Elementos del lenguaje Java



Salida de datos:

Para mostrar datos por pantalla en nuestros programas Java

En consola:

```
System.out.println("Nombre: " + nombre);  
System.out.println("Edad: " + edad);  
System.out.println("Edad en 10 años: " + (edad+10));
```

En cuadros de diálogo:

```
JOptionPane.showMessageDialog(null, "Nombre: " + nombre);  
JOptionPane.showMessageDialog(null, "Edad: " + edad);  
JOptionPane.showMessageDialog(null, "Edad en 10 años: " + (edad+10));
```



2.5 Traducción de un programa: compilación, enlace, ejecución y errores

Todo programa que no está en lenguaje máquina tiene que traducirse a lenguaje máquina antes de ser ejecutado.

Programa fuente:

Programa escrito en un lenguaje de programación que no es lenguaje máquina.

Programa objeto:

Programa en lenguaje máquina que puede ser ejecutado.



2.5 Traducción de un programa: compilación, enlace, ejecución y errores

Formas de traducir código fuente a código objeto:

Compilación:

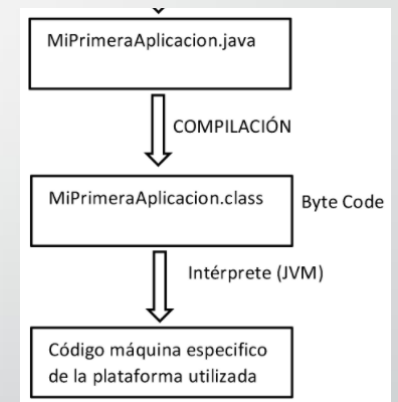
Un compilador traduce todo el código fuente en un código objeto y lo guarda para ser ejecutado.

- **se ejecuta más rápido**
- **el código objeto no requiere el compilador para ejecutarse**

Interpretación:

Un interprete traduce y ejecuta una a una, cada una de las instrucciones de un programa fuente.

- **se ejecuta menos rápido**
- **Se requiere el interprete para la ejecución.**



2.5 Traducción de un programa: compilación, enlace, ejecución y errores

En Java:

