



Glosario de términos

Fco. Javier Bohórquez Ogalla

Índice

1. Programa	5
2. Lenguaje de programación	5
3. Lenguaje de programación de alto nivel	5
4. Lenguaje de programación de bajo nivel	5
5. Lenguaje de programación compilados	5
6. Lenguaje de programación interpretados	6
7. Lenguaje de programación con tipado dinámico	6
8. Lenguaje de programación con tipado estático	6
9. Script	6
10. Interprete	6
11. Compilador	7
12. Operador	7
13. Expresión	7
14. Instrucción	7
15. Sentencia	7

16.Dato	8
17.Tipos de datos	8
18.Identificador	8
19.Variable	8
20.Función	8
21.Signatura	9
22.Procedimiento	9
23.Clase	9
24.Objeto	9
25.Argumento	10
26.Parámetro	10
27.Analizador léxico	10
28.Analizador sintáctico	10
29.Programación imperativa	11
30.Programación declarativa	11
31.Paradigma de programación	11
32.Constante	11

33.Código fuente	12
34.Árbol sintáctico	12
35.Gramática	12
36.Gramática libre de contexto	12
37.Biblioteca de programación	13
38.Biblioteca dinámica	13
39.Biblioteca estática	13
40.Módulo	13
41.Extensión o complemento	13
42.Token	14
43.Tipo lógico	14
44.Tipo numérico	14
45.Tipo array	14
46.Tipo cadena de caracteres	14
47.Tipo expresión regular	15
48.Comando	15

1. Programa

Secuencia de instrucciones que al ser ejecutadas por una computadora se corresponderán con la realización de una tarea específica. Para que las instrucciones que conforman un programa puedan ser ejecutadas por una determinada computadora deben presentarse en un formato legible por esta, generalmente binario, y deben estar dentro del conjunto de instrucciones que soporta.

2. Lenguaje de programación

Lenguaje formal generalmente usado para crear programas. Con un lenguaje de programación es posible expresar los algoritmos que determinan el comportamiento que debe llevar a cabo un determinado programa. Está formado por un conjunto de símbolos que representa el léxico y un conjunto de reglas sintácticas y semánticas.

3. Lenguaje de programación de alto nivel

Permiten expresar algoritmos de una forma abstracta, adecuada a la capacidad cognitiva humana. Son más cercanos al lenguaje humano que al lenguaje máquina. Un programa codificado en un lenguaje de alto nivel necesita ser procesado y transformado al conjunto de instrucciones que la computadora puede ejecutar.

4. Lenguaje de programación de bajo nivel

Son lenguajes cercanos al hardware, condicionados a la computadora. Derivan del conjunto de instrucciones soportados por la máquina, y van desde la representación binaria de estas hasta el uso de nemotécnicos.

5. Lenguaje de programación compilados

Se refiere a aquellos lenguajes de alto nivel tal que, para ser ejecutados, los programas codificados deben ser sometidos a un proceso de traducción a lenguajes de bajo nivel. Una vez sometido al proceso de compilación un programa puede ser ejecutado directamente por la computadora para la que fue compilada.

6. Lenguaje de programación interpretados

Se refiere a aquellos lenguajes de alto nivel tal que, para ser ejecutados, los programas codificados deben ser procesados por un intérprete que se encargará de obtener su representación a bajo nivel a la vez que lo ejecuta.

7. Lenguaje de programación con tipado dinámico

En los lenguajes de programación de tipado dinámico el tipo de los datos es determinado en tiempo de ejecución. Generalmente el tipo de dato es asociado al valor de una variable y no a la variable en si.

8. Lenguaje de programación con tipado estático

En los lenguajes de programación de tipado estático el tipo de los datos es determinado en tiempo de compilación. Generalmente el tipo de dato es asociado a la variable en el momento de su declaración.

9. Script

Programa simple, representado por una lista de comandos que serán ejecutados por un determinado programa o motor de scripts. Normalmente son utilizados para automatizar procesos. Algunos entornos que pueden ser automatizados mediante scripts incluye determinadas aplicaciones software, páginas webs, los shells del sistema operativo o sistemas embebidos.

10. Interprete

Programa que ejecuta directamente las sentencias escritas en un lenguaje de programación o de scripts, sin necesidad de compilar estas a un lenguaje de bajo nivel.

11. Compilador

Programa que traduce un programa escrito en un lenguaje de alto nivel a otro, generalmente de bajo nivel.

12. Operador

Símbolo matemático que indica que se debe llevar a cabo una operación determinada sobre un cierto número de operandos. El operador toma los elementos iniciales y los relaciona con otro elemento de un conjunto final que puede ser de la misma naturaleza o no.

13. Expresión

Secuencia de símbolos que pertenecen a un lenguaje formal. Las expresiones deben cumplir una serie de reglas determinadas por el lenguaje en las que están escritas, de forma que admiten una determinada interpretación dentro del lenguaje y cuya evaluación la atribuirá de valor. En los lenguajes de programación una expresión suele ser una combinación de constantes, operadores, funciones y demás recursos del lenguajes.

14. Instrucción

Secuencia de bits que el procesador es capaz de interpretar y ejecutar. Las instrucciones que un determinado procesador es capaz reconocer viene determinada por el conjunto de instrucciones del mismo, dado en el momento de su fabricación y según la arquitectura con la que fue diseñado. También son consideradas instrucciones las representaciones de estas en lenguajes de nemotécnicos como ensamblador.

15. Sentencia

Unidad con valor semántico a partir de la cual se construye un lenguaje de alto nivel. Son para los lenguajes de alto nivel lo que las instrucciones los son para los de bajo nivel. Generalmente se componen de expresiones u otras construcciones propias del lenguaje.

16. Dato

Representan la unidad mínima de información. Son valores que los programas manipulan para construir la solución al problema que pretenden resolver.

17. Tipos de datos

Según la naturaleza de los datos estos pueden quedar organizados en tipos. Un lenguaje de programación generalmente opera sobre unos tipos de datos predefinidos, aunque ofrecen la capacidad de definir tipos de datos más complejos a partir de estos.

18. Identificador

Elemento textual o símbolo que es parte del léxico de un lenguaje y que nombran entidades del mismo como variables, constantes, funciones...

19. Variable

En el contexto de la programación una variable representa un espacio de almacenaje que contiene un valor (conocido o desconocido) que es asociado a un identificador. El valor guardado por una variable puede cambiar en el tiempo de ejecución del programa. El tipo de dato que puede almacenar una variable puede estar ligado a la variable, por lo que esta sólo podrá almacenar valores de un tipo determinado, o estar asociada al valor en sí por lo que la variable podrá almacenar valores de distintos tipos.

20. Función

En el contexto de la programación una función representa un conjunto de sentencias o instrucciones con un determinado propósito. Las funciones se ejecutan al ser llamadas desde otras funciones o procedimientos del programa, o incluso desde sí misma (funciones recursivas). Las funciones pueden recibir datos desde el punto desde el que son llamadas por medio de los denominados parámetros de la función. Las funciones generalmente están diseñadas para devolver un determinado valor fruto de la ejecución del algoritmo que codifican o encierran. Las funciones son un elemento de suma importancia dentro

de la programación funcional, pero son también muy utilizadas en otros paradigmas de programación, tanto declarativos como imperativos.

21. Signatura

La signatura de una función o método define el nombre o identificador del mismo, así como los parámetros de los que dispone. En algunos lenguajes puede incluir el tipo de dato de los parámetros y el tipo que es devuelto.

22. Procedimiento

Representa un conjunto de actividades, eventos o tareas que son ejecutadas para llevar a cabo un determinado propósito. Estas son codificadas mediante sentencias, llamadas a funciones u otros recursos del lenguaje.

23. Clase

En el contexto de la programación, y más concretamente dentro de la programación orientada a objetos, una clase es una estructura que define un estado por medio de variables, denominadas atributos, y un conjunto de operaciones que encapsulan un determinado comportamiento, denominados métodos. Las clases representan una plantilla o modelo a partir de la cual se crean objetos o instancias pertenecientes a la misma.

24. Objeto

En el contexto de la programación, y más concretamente dentro de la programación orientada a objetos, un objeto es una estructura que encapsula un determinado estado (atributos) y a la cual se le puede aplicar una serie de operaciones (métodos). Generalmente estos son creados mediante la instanciación de una clase de objeto, que define una serie de objetos con un comportamiento y estados afines.

25. Argumento

Los argumentos son parte de la llamada a una función o procedimiento. Un argumento representa el valor que es asociado a un determinado parámetro cuando es llamado. La definición puede extenderse a los valores que son pasados a un programa cuando este es ejecutado. Generalmente los argumentos son asociados a los parámetros de una forma posicional.

26. Parámetro

Los parámetros son parte de la signatura de una función o procedimiento. Un parámetro representa un valor que una función o procedimiento espera que sea transferido cuando son llamados. En algunos lenguajes de programación, determinados parámetros pueden presentar valores por defectos, que son los que se tomarán si no están presente en la llamada.

27. Analizador léxico

Un analizador léxico o scanner es un componente software cuya implementación normalmente se corresponde con un autómata finito. Este se encarga de determinar si una determinada cadena pertenece o no al conjunto de cadenas que conforma el léxico del lenguaje. Esta pieza software recibe como entrada un contenido fuente escrito generalmente como cadenas de caracteres, aunque puede soportar otros tipos de codificaciones (audio, imágenes...), y devuelve un conjunto de tokens correspondientes.

28. Analizador sintáctico

Un analizador sintáctico o parser es un componente software cuya implementación normalmente se corresponde con un autómata de pila (si la gramática es libre de contexto). Este se encarga, ayudado generalmente de un analizador léxico, de tomar un contenido fuente escrito generalmente como cadena de caracteres (aunque puede soportar otros tipos de codificaciones), y construir una estructura más fácil de analizar y procesar (normalmente árboles). El resultado de llevar a cabo un análisis sintáctico normalmente es un árbol de derivación denominado árbol sintáctico.

29. Programación imperativa

El programa se ve como una entidad que presenta un estado y una serie de sentencias u operaciones que hacen que dicho estado cambie. Este tipo de programación es cercana a la máquina, ya que la implementación de la mayoría de computadores a nivel hardware es imperativa.

30. Programación declarativa

Los lenguajes declarativos utilizan construcciones matemáticas para describir el problema y así obtener la solución.

En los lenguajes declarativos puros se cumple una transparencia referencial en todo el sistema por lo que se evitan efectos colaterales. Además no existen las asignaciones destructivas. Esto marca una diferencia con los lenguajes imperativos y es que las funciones declarativas no pueden depender o cambiar el estado del programa. Los lenguajes multiparadigma pueden ofrecer estructuras que garanticen estos principios.

31. Paradigma de programación

Representa un estilo fundamental de programación, sirve como una forma de construir estructuras y elementos de los programas. Las capacidades y estilos de muchos lenguajes de programación son definidos para soportar determinados paradigmas de programación. Algunos lenguajes son diseñados para seguir un único paradigma, mientras que otros persiguen el soporte para varios de estos.

32. Constante

En el ámbito de la programación una constante representa un valor que no varía en el tiempo, por lo que una vez declarado solo podrá leerse y no modificarse. El valor constante puede estar asociado a un identificador o nombre que se usará para referenciarlo.

33. Código fuente

El código fuente representa las instrucciones o sentencias que conforman el programa. Generalmente estas se presentarán en un formato de cadena de texto, aunque determinados sistemas pueden ser presentados en otros formatos como imágenes, audio ...

34. Árbol sintáctico

Árbol de derivación producto del análisis sintáctico. La raíz y demás nodos no hojas se componen de símbolos no terminales, mientras que los nodos hojas se componen únicamente de símbolos terminales

35. Gramática

Estructura que representa unas reglas de formación que define cadenas o frases que pertenecen a un determinado lenguaje natural o formal.

Una gramática es un cuádrupla $G = (VT, VN, S, P)$, donde:

VT : Conjunto de símbolos terminales.

VN : Conjunto de símbolos no terminales.

S : Símbolo inicial de la gramática, $S \in VN$.

P : Reglas de derivación.

Los tipos de gramáticas generalmente vienen determinadas por los tipos de reglas de derivación que las componen.

36. Gramática libre de contexto

Son gramáticas cuyas reglas de derivación no dependen de un contexto. Estas son de la forma $V \rightarrow w$ donde V es un símbolo no terminal y w es una cadena de terminales y no terminales. Las gramáticas libres de contexto originan lenguajes libres de contexto que pueden ser implementados mediante autómatas de pilas.

37. Biblioteca de programación

Una Biblioteca es un conjunto de funciones o componentes de programación agrupados y encapsulados, que se encuentran relacionados y que comparten características afines. Su principal función es la reutilización de funcionalidades entre programas. En los lenguajes compilados se pueden distinguir entre bibliotecas dinámicas o estáticas.

38. Biblioteca dinámica

Una Biblioteca de programación es dinámica si se encuentra ya compilada. El programa carga y hace uso de esta en tiempo de ejecución.

39. Biblioteca estática

Una Biblioteca de programación es estática si se encuentran en un lenguaje de alto nivel. Estas son añadidas y compiladas juntos al programa.

40. Módulo

Parte de un software informático que lleva a cabo una función específica dentro del conjunto de tareas que esta realiza. Generalmente los módulos de un programa se encuentran organizados jerárquicamente según el nivel de abstracción que presentan y el objetivo que cumplen.

41. Extensión o complemento

Sistema software que se relaciona con otro para extender o añadir funcionalidades de este. El grado de dependencia de las extensiones con la aplicación base es muy alto y en un solo sentido.

42. Token

Elemento léxico con cierto valor para un determinado lenguaje de programación. Normalmente se corresponde con una cadena de caracteres que se puede corresponder con una palabra reservada, un identificador, un número... Un token puede contener un valor.

43. Tipo lógico

Un valor lógico, también denominados booleanos, representa un valor verdadero o falso. Un valor lógico es equivalente a un valor binario 0 ó 1.

44. Tipo numérico

Un valor numérico, también denominados aritméticos, representa un valor entero o real. Un valor entero comprende los números positivos, negativos y el cero. Un valor real posee una parte entera y otra decimal.

45. Tipo array

Un array o vector representa un conjunto ordenado de valores o elementos del mismo que se encuentran posicionados en memoria de forma contigua. En muchos lenguajes esta definición suele extenderse a listas de elementos que pueden tener diferente tipo.

46. Tipo cadena de caracteres

Una cadena de caracteres es una secuencia de caracteres, que comprende signos, símbolos, letras o números. En el ámbito de la programación se utiliza normalmente como un tipo de dato compuesto, representado mediante un array cuyos elementos son los caracteres que componen la cadena.

47. Tipo expresión regular

Cadena de caracteres que representa un lenguaje regular, normalmente conforman un patrón y son utilizadas para buscar o sustituir cadenas dentro de otras.

48. Comando

Instrucción u orden que el usuario proporciona a un sistema informático. El sistema indica al usuario que espera un comando por medio de una cadena de caracteres denominada prompt.