

Sintaxis básica

Cuando desarrollamos un programa en Java es muy importante tener en cuenta los siguientes puntos:

- **Sensibilidad a mayúsculas** - Java es sensible a mayúsculas, lo que significa que si poseemos un identificador llamado Hola y hola tendrían valores diferentes.
- **Nombres de las clases** - Para todos los nombres de clases la primera letra debe estar en mayúsculas. Si quieres usar varias palabras para formar un nombre, la primera letra de cada palabra interior debe estar en mayúsculas.

```
class MiPrimeraClase
```

- **Nombres de los métodos** - Todos los nombres de los métodos deben comenzar con una letra minúscula. Si se quiere usar varias palabras para formar el nombre de un método, la primera letra de cada palabra interior debe estar en mayúsculas .

```
public void miPrimerMetodo()
```

- **Nombre de archivo del programa** - El nombre del archivo de programa debe coincidir exactamente con el nombre de la clase.

Al guardar un archivo, debemos guardarlo con el nombre de clase (Recuerda que Java distingue entre mayúsculas y minúsculas) y añadir '.java' al final del nombre (si el nombre del archivo y el nombre de clase no coinciden el programa no compilará).

Ejemplo: Supongamos que 'MiPrimerPrograma' es el nombre de la clase. Luego el archivo debe ser guardado como 'MiPrimerPrograma.java'

- **public static void main(String args[])** - la ejecución de un programa en Java se inicia desde el método main(), por lo cual es una parte obligatoria del desarrollo.

IDENTIFICADORES

Todos los componentes en Java requieren nombres. Los nombres usados para las clases, variables y métodos se denominan identificadores.

Existen varios puntos que debemos recordar acerca de los identificadores:

- Todos los identificadores deben comenzar con una letra (A a Z o de la a a z), el carácter de dólar (\$) o un guión bajo (_).
- Una palabra clave no se puede utilizar como un identificador .
- Los identificadores distinguen entre mayúsculas y minúsculas.
- Ejemplos de identificadores legales: edad, \$salario, _valor, __1_valor.
- Ejemplos de identificadores ilegales : 123abc , -salario.

MODIFICADORES

Al igual que otros lenguajes de programación, en Java es posible modificar las clases, métodos, etc..., mediante el uso de modificadores. Existen dos categorías de modificadores:

- De acceso: default, public , protected, private.
- De no acceso: final, abstract.

VARIABLES

En Java existen los siguientes tipos de variables:

- Variables locales.
- Variables de clase (variables estáticas).
- Variables de instancia (variables no estáticas).

ARRAYS

Los arrays son objetos que almacenan múltiples variables del mismo tipo. Sin embargo, un array en sí es un objeto.

COMENTARIOS

Java soporta una o varias líneas de comentarios. Es muy similar a los comentarios que podemos encontrar en C y C++. Todos los caracteres disponibles dentro de cualquier comentario son ignorados por el compilador.

```
1  public class MiPrimerPrograma{
2
3      /* Este es mi primer programa
4       * Esto va a imprimir "Hola Mundo"
5       * Un ejemplo de un comentario multi-línea
6       */
7
8      public static void main(String []args){
9          // Un ejemplo de un comentario de una línea
10         /* Este también es un ejemplo de un comentario de una línea */
11         System.out.println("Hola Mundo");
12     }
13 }
```

HERENCIA

En Java, las clases pueden ser derivados de clases. Básicamente, si necesitamos crear una nueva clase y tenemos una clase que tiene una parte del código que necesitamos, entonces es posible derivar la nueva clase a partir del código ya existente.

Este concepto permite reutilizar los campos y métodos de la clase existente sin tener que volver a escribir el código de una nueva clase. En este escenario la clase existente se llama la superclase y la clase derivada se llama la subclase.

PALABRAS CLAVE

La siguiente lista muestra las palabras reservadas de Java. Estas palabras reservadas no se pueden utilizar como constante o variable o cualquier otro nombre de identificador.

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

INTERFACES

En Java, una interfaz se puede definir como un contrato entre los objetos sobre la forma como se van a comunicar entre sí. Las interfaces juegan un papel fundamental cuando se trata de el concepto de herencia.

Una interfaz define los métodos que una clase derivada (subclase) debe utilizar. Pero la puesta en práctica de los métodos es totalmente de la subclase.

CLASES

Una clase es un colección de miembros dato y métodos que define un objeto específico. Es la construcción fundamental del lenguaje Java. Una clase puede ser definida por el usuario o por uno de los paquetes incorporados a Java y se declara con la palabra reservada "class". El convenio utilizado para nombrar las clases utilizar una letra mayúscula para la primera letra del nombre de la clase.

Los datos o variables, definidas dentro de una clase se denomina Variables de instancia. El código está contenido dentro de métodos .de modo colectivo ,los métodos y las variables definidas dentro de una clase se denominan miembros de la clase. Las variables definidas dentro de una clase se llaman variables de instancia ya que cada instancia de la clase (es decir, cada objeto de la clase) contiene su propia copia de estas variables. Cada clase define un nuevo tipo de datos.

TABLAS DE VERDAD EN JAVA

Veamos cómo elaborar una tabla de verdad.