

# CURSO DE PROGRAMACIÓN CON JAVA

## 1. SINTAXIS BÁSICA DE JAVA



Por el experto: Ing. Ubaldo Acosta



[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

Hola, te saluda Ubaldo Acosta. Bienvenidos nuevamente a este curso de Programación con Java.

En esta primera lección vamos a revisar la sintaxis básica, la cual estudiamos a detalle en el curso anterior, esto es sólo un breve repaso para que comencemos a calentar motores. Veremos desde la definición básica de una Clase, hasta como instanciar objetos en Java, entre ejemplos de sintaxis más.

Así, que si estás listo, nosotros también. Comencemos de inmediato.

## ESTRUCTURA DE UNA CLASE

### Sintaxis básica de la estructura de una Clase:

```

package <nombre_paquete>;

import <nombre_paquetes>;

public class NombreClase {

    <atributos>;

    <Constructores>;

    <Métodos>;

}
  
```



### CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

Esta es una breve revisión de los elementos de una clase.

En primer lugar especificamos el paquete al cual pertenece la clase. Un paquete básicamente es como una carpeta que agrupa varias clases que tienen algo en común, normalmente se agrupan las clases por su funcionalidad.

Posteriormente, si es necesario, realizamos los import de las clases que vayamos a utilizar. Recordemos que hay import de clases, o también import static los cuales importan atributos o métodos estáticos para ser utilizados directamente en nuestro código .

Después indicamos el nombre de nuestra clase. Recordemos que en un archivo sólo puede declararse una clase de tipo public, y esta debe llamarse exactamente igual que el nombre del archivo que tiene extensión .java, sin embargo pueden existir más clases en un mismo archivo, las cuales ya no son publicas, sino solo clases definidas dentro del mismo archivo.

Seguido al nombre de la clase, definimos los atributos de la clase. El orden de los elementos no afecta a la clase, ya que si recordamos el diagrama de donas de un objeto, podemos entender que los atributos y métodos se pueden definir en cualquier parte dentro de la clase, pero como una buena práctica se recomienda hacerlo en el orden mostrado. Los atributos pueden ser de cualquier tipo de dato que ya hemos estudiado previamente, además de contener varios modificadores, como los modificadores de acceso, o static, final entre varios más que estudiaremos más adelante.

Posterior a los atributos se recomienda declarar los constructores, de igual manera el orden no afecta, pero es una buena práctica definirlos después de los atributos de nuestra clase. Pueden existir varios constructores, tanto privados como públicos, con varios argumentos o sin argumentos. Si no definimos un constructor, entonces el compilador en automático agrega el constructor vacío para que se puedan crear objetos de la clase que definimos. Si se define algún constructor distinto al vacío, entonces el compilador ya no agrega el constructor vacío a nuestra clase, y se deberá utilizar alguno de los constructores definidos para crear un objeto de la clase definida.

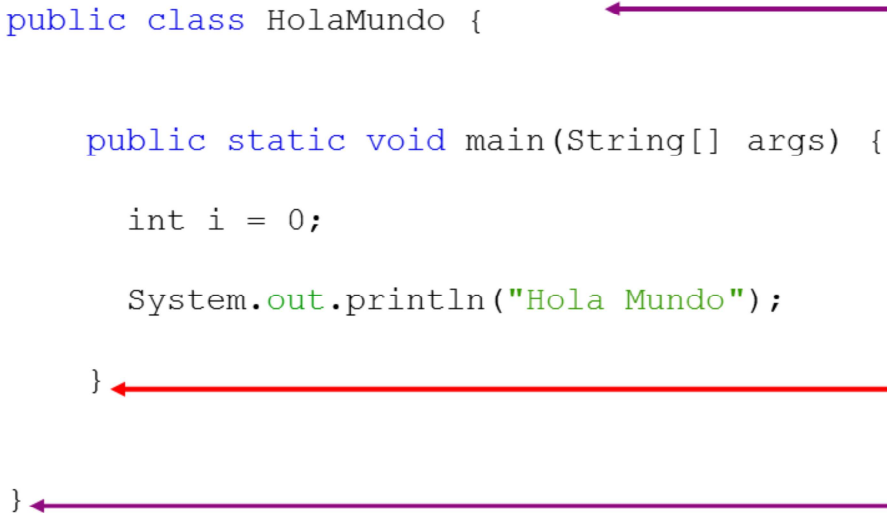
Finalmente definimos los métodos de nuestra clase, pueden existir de varios tipos, como son simples métodos de tipo get o set para las propiedades de la clase, o métodos estáticos, públicos, privados, etc. Todo depende de la funcionalidad que necesitemos del método definido. Esto se conoce como la interfaz del método, y no tienen nada que ver con el tema de interfaces que veremos más adelante. La interfaz de una clase tiene que ver con los métodos que tenemos disponibles para ser utilizados en una clase.

Con esto a grandes rasgos tenemos un resumen de la creación de clases en Java. Recordemos que hay varios temas más, como el concepto de herencia y más temas, pero dejemos simple la estructura básica de la clase para comenzar a estudiar otros temas, y en su momento retomaremos cada uno de los temas estudiados en el curso anterior, así como los nuevos temas incluidos en este curso.

## BLOQUES DE CÓDIGO

### Bloques de código en Java:

```
public class HolaMundo {  
  
    public static void main(String[] args) {  
        int i = 0;  
        System.out.println("Hola Mundo");  
    }  
}
```



### CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

Las clases, así como los métodos tienen un inicio y un fin, el símbolo que se utiliza para marcar el inicio y el fin son las llaves { }. Esto se conoce como bloque de código, incluso existen bloques de código anónimos que no tienen ningún nombre, esto lo estudiaremos más adelante.

Las variables definidas dentro de estos bloques de código tienen precisamente la duración hasta donde termina el bloque de código donde se declara.

Por ejemplo si una variable se declara dentro del método main, únicamente existirá durante la ejecución de este método, y terminará de existir cuando este bloque de código termine, es decir, cuando la ejecución del método main termine.

El compilador de Java ignora cualquier espacio o salto de línea, de tal manera que podemos utilizar precisamente esta característica para hacer y escribir un código más legible, conforme avancemos veremos varias buenas prácticas para escribir el código, ya que el compilador no indicará que hay un error, sino que nosotros deberemos utilizar estas prácticas para que nuestro código sea más legible y entendible tanto para nosotros como para el equipo de trabajo del proyecto en cuestión.

Este es sólo un breve repaso del tema de alcance de variables estudiado en el curso de Fundamentos de Java.



## CONVENCIONES EN JAVA

### Ejemplo de Convenciones en Java:

```
public class Persona{
    //Atributos
    private String nombre;
    private Date fecNacimiento;
    private String domicilios[];
    private final int MAX_DOMICILIOS = 3;

    public void agregarDomicilio(){
        if(domiciliosAgregados < MAX_DOMICILIOS){
            //agregar domicilio
        }
    }
}
```

Las clases son sustantivos, comienzan con mayúscula y siguen notación de camello

Una constante se escribe en mayúsculas.

Los métodos son verbos

Los bloques de código se indentan como se muestra

### CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

Veamos ahora el tema de convenciones. Una convención es un acuerdo, por lo que debemos entender que el compilador no arrojará ningún error o nos indicará que hace falta corregir o cambiar algo. Por ello esto lo debemos tomar como buenas prácticas y es finalmente el equipo de trabajo el que definirá las convenciones a utilizar en el código que creemos. Sin embargo estas son algunas convenciones generales que les sugerimos seguir con el objetivo que su código sea más legible y entendible tanto para nosotros como para nuestro equipo de trabajo del proyecto respectivo.

- Las clases son sustantivos, un sustantivo es básicamente el nombre que le damos a una persona o cosa. Puede existir físicamente o ser algo abstracto, como un concepto creado. Por ello el nombre que le demos a nuestras clases serán sustantivos, y el nombre deberá seguir algunas convenciones. La primera letra debe estar escrita en mayúsculas, y posteriormente utilizar la notación de camello. Esta notación quiere decir que si nuestra clase utiliza dos nombres, debemos unirlos, sin separación y cada palabra debe iniciar en mayúsculas, por ejemplo. TarjetaCredito.
- Los métodos deben ser verbos, y la primer letra debe ser escrita en minúscula, y posteriormente utilizar la notación de camello, por ejemplo: retirarDinero(), o cambiarEstado(), etc.
- Las variables deben ser breves pero con significado sustancial, por ejemplo fecNacimiento, sin embargo es preferible no abreviar demasiado sus nombres de tal modo que con solo leer el código podamos entender el uso de la variable definida. Variables con una sola letra deben ser evitadas, excepto cuando sean variables que se ejecutan en un bloque de código pequeño y de manera temporal, como puede ser un ciclo for al definir la variable i que sirve de contador para la iteración de este ciclo.
- Las constantes, que son las variables definidas con la palabra final, deben estar escritas en mayúsculas y si se utiliza más de una palabra cada palabra debe estar separada por un guio bajo.
- La indentación significa el uso de espacios en blanco o saltos de línea con el objetivo de hacer nuestro código más sencillo de leer y por consiguiente más sencillo de hacer modificaciones. Por ello los métodos se indentarán según la lámina descrita, es decir, se abre la llave del bloque de código al terminar el método, y al finalizar la última línea del método se agregará en la línea siguiente el cierre de la llave del método respectivo.

Para más información de las convenciones en Java, se creó un documento, el cual puede ver en el siguiente link:  
<http://icursos.net/cursos/ProgramacionJava/Leccion02/doc/convencionesJava.pdf>

CURSO ONLINE

# PROGRAMACIÓN CON JAVA

Por: Ing. Ubaldo Acosta



CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

En Global Mentoring promovemos la Pasión por la Tecnología Java. Te invitamos a visitar nuestro sitio Web donde encontrarás cursos Java Online desde Niveles Básicos, Intermedios y Avanzados, y así te conviertas en un experto programador Java.

A continuación te presentamos nuestro listado de cursos:

- ✔ Lógica de Programación
- ✔ Fundamentos de Java
- ✔ Programación con Java
- ✔ Java con JDBC
- ✔ HTML, CSS y JavaScript
- ✔ Servlets y JSP's
- ✔ Struts Framework
- ✔ Hibernate Framework
- ✔ Spring Framework
- ✔ JavaServer Faces
- ✔ Java EE (EJB, JPA y Web Services)
- ✔ JBoss Administration
- ✔ Android con Java
- ✔ HTML5 y CSS3

#### Datos de Contacto:

Sitio Web: [www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

Email: [informes@globalmentoring.com.mx](mailto:informes@globalmentoring.com.mx)

