

CURSO DE FUNDAMENTOS DE JAVA

5. CLASES Y OBJETOS



Por el experto: Ing. Ubaldo Acosta



CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

Hola, te saluda Ubaldo Acosta. Bienvenida o bienvenido nuevamente. Espero que estés listo para comenzar con esta quinta lección.

Vamos a estudiar el tema de Clases y Objetos, esto es la base de la programación orientada a objetos.

¿Estás listo? Ok, ¡Vamos!

INTRODUCCIÓN A LAS CLASES Y OBJETOS EN JAVA

UNA CLASE ES UN PLANTILLA

POSEE UN NOMBRE



CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

En Java el concepto de Clases es fundamental para poder manejar este lenguaje de manera adecuada, ya que es el fundamento de este increíble lenguaje de programación.

Una clase define la naturaleza de un objeto, por lo tanto una clase provee las bases de la programación orientada a objetos. Así que cualquier concepto que se vaya a implementar en Java está envuelto en una clase. En este capítulo veremos como crear una clase y a su vez cómo podemos crear objetos a partir de nuestra clase.

Hemos utilizado las clases desde el inicio del curso, pero hasta ahora es que explicaremos este concepto. Hasta el momento las clases que hemos creado han sido creadas con el único objetivo de encapsular el método main y poder ejecutar nuestros programas. Sin embargo empezaremos a ver en qué consiste una clase y cómo podemos beneficiarnos de este paradigma de Programación Orientado a Objetos, también conocido como POO.

Quizá uno de los puntos más importantes a entender es que una clase genera un nuevo tipo de datos en Java. Una vez definido este tipo de datos, entonces lo podemos utilizar para crear objetos del tipo previamente definido. Por lo tanto una clase es una plantilla para poder crear objetos. Debido a que un objeto es una instancia de una clase, normalmente utilizaremos “objeto” o “instancia de una clase” de manera equivalente.

Por ejemplo, podemos observar en la figura una plantilla de una Persona, esta persona puede tener atributos (nombre, género, ocupación, etc), y métodos para ejecutar las acciones que puede realizar esta persona (comer, dormir, brincar, etc). A continuación veremos como definir estos elementos de una clase a más detalle.

TIPOS EN JAVA

Tipos de Datos en Java

Tipos Primitivos

Tipos Enteros

Tipos Flotantes

Tipos Referenciados (Tipo Object)

Clases

Interfaces

Arreglos

CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

Recordemos rápidamente que los tipos en Java, además de tener tipos primitivos, también existen tipos referenciados. Estos tipos pueden provenir de Clases, Interfaces o Arreglos. En esta lección estudiaremos los tipos Object que son instancia de una clase, y posteriormente estudiaremos los tipos Interface y Arreglos.

FORMA GENERAL DE UNA CLASE EN JAVA

Nombre de la clase debe terminar con extensión .java:

```
class NombreDeLaClase {

    tipoDato variableDeInstancia1;
    tipoDato variableDeInstancia2;
    //Más variables de instancia...

    tipoDato nombreDelMetodo1(argumentos) {
        //Cuerpo del método
    }

    tipoDato nombreDelMetodo1(argumentos) {
        //Cuerpo del método
    }
    //Más métodos...
}
```

www.globalmentoring.com.mx

En la lámina podemos ver la estructura general de una clase en Java. Como hemos comentado una Clase es una plantilla o molde, de la cual podremos generar objetos (instancias) para que puedan ser finalmente utilizados estos nuevos tipos de datos en nuestros programas.

Existen mas variantes, pero en esta lámina vamos a simplificar la creación de una clase, para que tengamos una idea más concreta de lo que es y cómo podemos empezar a trabajar con ellas. Vamos a ver los pasos generales para crear una clase:

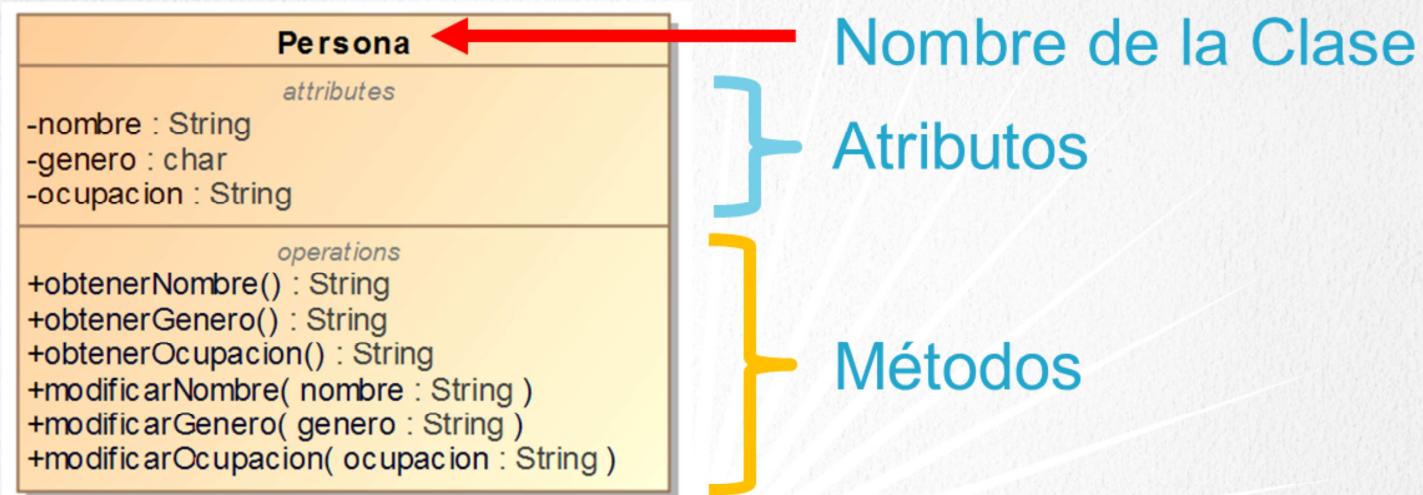
- 1) Definir el nombre de la clase, anteponiendo la palabra reservada class. Esta clase se debe guardar en un archivo con el mismo nombre (recordar que Java es sensible a mayúsculas y minúsculas) con la extensión .java
- 2) Definir los atributos o variables de nuestra clase. A esto se le conoce como variables de instancia de nuestra clase.
- 3) Definir los métodos de nuestra clase. Los métodos son los que realmente contienen el código de nuestra clase, es decir, la funcionalidad y objetivo de ser de nuestra clase.

Es importante decir que la definición de una clase precisamente es definir nuestra plantilla, a partir de la cual crearemos objetos que son con los que finalmente trabajaremos en nuestra aplicación.

En muchas ocasiones cuando estamos iniciando el proceso de creación de una clase suele ser algo complicado de entender, ya que debemos agregar variables y métodos a nuestra clase, de la cual aún no sabemos exactamente su funcionalidad, entonces la pregunta es: ¿Debemos definir todos los atributos y/o métodos desde un inicio a nuestra clase, o debemos esperar a que las necesidades o requerimientos vayan complementando el código de la misma?

La respuesta a esta pregunta depende en muchas ocasiones de la metodología de desarrollo de software que estemos utilizando, pero en general, agregaremos los métodos que surjan del análisis de nuestro sistema. Así que posiblemente la clase tendrá más métodos y atributos que vayamos a utilizar en un inicio, pero nuestra clase o plantilla estará lista para soportar futuros cambios en nuestro sistema. Por ello es que en ocasiones encontraremos atributos o métodos en nuestras clases que son poco o nunca utilizados al momento de crear los objetos de dichas plantillas. Más adelante veremos cómo crear objetos a partir de nuestra clase.

DIAGRAMA GENERAL DE UNA CLASE EN JAVA



CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

En la lámina podemos ver la estructura general de una clase en Java. Como hemos comentado una Clase es una plantilla o molde, de la cual podremos generar objetos (instancias) para que puedan ser finalmente utilizados estos nuevos tipos de datos en nuestros programas.

Existen mas variantes, pero en esta lámina vamos a simplificar la creación de una clase, para que tengamos una idea más concreta de lo que es y cómo podemos empezar a trabajar con ellas. Vamos a ver los pasos generales para crear una clase:

- 1) Definir el nombre de la clase, anteponiendo la palabra reservada class
- 2) Definir los atributos o variables de nuestra clase. A esto se le conoce como variables de instancia de nuestra clase.
- 3) Definir los métodos de nuestra clase

Cada uno de estos puntos son los que estaremos estudiando y profundizando en las siguientes lecciones, desde la creación de clases, definición de atributos de una clase, así como la creación de métodos en una clase.

EJEMPLO DE UNA CLASE: PERSONA

Clase Persona:

```
//Nombre de la clase
class Persona {

    //Atributos de la clase
    String nombre;
    String apellidoPaterno;
    String apellidoMaterno;

}
```

CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

Comencemos el estudio de las clases, con la definición de una clase muy simple. Podemos observar la definición de la clase Persona, la cual contiene 3 atributos de tipo String y sin ningún método.

Como hemos comentado, al definir esta clase, estamos creando un nuevo tipo de dato en Java. En este caso el nuevo tipo de dato se llama Persona. Usaremos este nombre para crear objetos de tipo Persona. Es importante recordar que al definir una clase, sólo estamos definiendo la plantilla con la que vamos a trabajar, pero para poder usar esta plantilla necesitamos crear objetos a partir de esta plantilla. Definir la clase no crea un objeto en automático, tenemos que crearlos. Esto es lo que veremos a continuación.

OBJETOS EN JAVA



CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

Un objeto es la instancia de una clase, es decir que a partir de nuestra plantilla, es posible crear objetos en concreto con los cuales podremos trabajar, es decir, cambiar sus valores y llamar a sus métodos.

Más adelante veremos que hay casos en los que podremos trabajar directamente con la clase, a esto se le llama contexto estático, pero de momento estaremos estudiando el contexto dinámico, es decir, necesitamos crear una instancia de una clase para poder trabajar con ellas.

Como podemos observar en la figura, tenemos la clase Persona, y a partir de esta plantilla podemos crear objetos concretos para poder trabajar con sus características y atributos. Por ejemplo hemos creado dos objetos de tipo Persona, y cada uno tiene características similares pero con valores distintos.

Las características son por ejemplo: Nombre, Apellido, Color, Género, escolaridad, etc.

Y cada uno de los objetos se les asigna estas características, de esta manera la clase nos sirve sólo para indicar que características tendrán los objetos que vamos a crear. Veamos ahora la sintaxis para crear un objeto.

CREACIÓN DE UN OBJETOS EN JAVA

Creación de un objeto Persona:

```
public class PersonaPrueba {
    public static void main(String args[]) {
        //Creacion de un objeto
        Persona p1 = new Persona();

        //Modificar valores
        p1.nombre = "Armando";
        p1.apellidoPaterno = "Esparza";
        p1.apellidoMaterno = "Lara";
    }
}
```

CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

Para definir un objeto vamos a utilizar la palabra reservada new. Esta palabra en Java significa en términos simples que vamos a crear una nueva variable del tipo Persona. Para crear un objeto vamos a utilizar una línea de código como sigue:

```
Persona p1 = new Persona();
```

Después de que esta línea se ejecuta, la variable p1 podrá acceder a un objeto de tipo Persona. Cada vez que se manda llamar la palabra new se creará un nuevo objeto de tipo Persona, y contendrá sus propios valores. Por ejemplo:

```
Persona p2 = new Persona();
```

Esta nueva variable p2, apunta a un nuevo objeto de tipo Persona, y por lo tanto contendrá valores distintos a la variable p1. Así, cada nuevo objeto contendrá valores distintos en los atributos de nombre, apellidoPaterno y apellidoMaterno. La sintaxis new Persona() hace referencia al nombre de la clase de la cual queremos crear una nueva instancia, es decir, un objeto. Además, estamos haciendo uso del concepto de constructor de la Clase, este concepto lo estudiaremos más adelante, aquí solo debemos notar que para llamar al constructor de la clase se pone el nombre de la clase y posteriormente se utilizan paréntesis, en este caso un constructor sin argumentos, pero veremos posteriormente que podemos enviar argumentos al momento de crear un objeto.

El nombre de la variable puede ser cualquiera que elijamos, pero no debe tener caracteres especiales, o espacios. Para acceder a los atributos de nuestras clases vamos a utilizar el operador . A través de este operador se liga el nombre del objeto con el nombre del atributo de la clase. Por ejemplo, para asignar el valor de Armando al atributo nombre del objeto p1, sería como sigue:

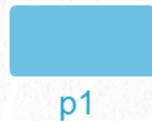
```
p1.nombre = "Armando";
```

En lecciones posteriores veremos en términos más técnicos lo que hace la palabra new, pero para esta lección es suficiente con saber que nos permite crear nuevos objetos.

MÁS DE LA PALABRA NEW

Sentencias

```
//Declaración de una variable
Persona p1;
```



```
//Creacion de un objeto
Persona p2 = new Persona();
```



Efecto en la Memoria

CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

La palabra new solicita espacio de memoria de manera dinámica para poder almacenar un objeto. Con la sintaxis generalmente como sigue:

```
variableQueAlmacenaReferencia = new TipoClaseInstanciar();
```

En la figura podemos observar que en primer lugar declaramos una variable de tipo de la clase Persona, llamada p1. Podemos observar que el efecto que esto tiene en la memoria Java es que se crea espacio para esta variable llamada p1. Pero esta variable aún no tiene ningún valor asignado, ya que sólo se declaró la variable.

En la segunda sentencia, podemos observar, que además de declarar una variable de tipo Persona llamada p2, se le asigna la referencia de un objeto de tipo Persona. Para la creación de este objeto de tipo persona se utiliza la palabra new, seguida del nombre de la clase de la cual queremos crear el objeto y finalmente los paréntesis es el tipo de constructor que vamos a utilizar para la creación de este objeto.

El concepto de constructor lo estudiaremos a detalle posteriormente, pero solo pensemos en el constructor como la forma en que queremos crear nuestro objeto de manera inicial, podemos crear por ejemplo un objeto de tipo persona ya con valores como son nombre, apellido, etc, o como en este caso, podemos simplemente crear el objeto de tipo Persona, sin proporcionar los valores de nombre y apellido y posteriormente podremos cambiar y/o asignar estos valores a los atributos de nuestro objeto de tipo Persona.

Debemos aclarar que los tipos primitivos como int, long, etc, no son objetos, por lo tanto para declarar variables de estos tipos no se necesita utilizar la palabra reservada new.

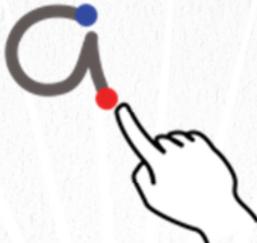
Recordemos que la palabra new lo que hace internamente es solicitar espacio de memoria para la creación de un nuevo objeto, tantos objetos como necesitemos, sin embargo, la memoria es finita, y por lo tanto podríamos recibir un error al momento de crear demasiados objetos en tiempo de ejecución de nuestro programa. En temas posteriores veremos el manejo de errores y como tratar los más comunes, como la falta de memoria en Java.

En conclusión, una clase define un nuevo tipo, y un objeto es una instancia de este nuevo tipo que ya se ha creado previamente. Podemos crear tantos objetos del tipo creado como necesitemos, esto con ayuda de la palabra new.

USO DE LA CLASE STRING

**NO ES UN
TIPO
PRIMITIVO**

**TRATAMIENTO
ESPECIAL**



**NO SE
NECESITA LA
PALABRA NEW**

Ej. String saludo = "Hola Mundo";

CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

Si recordamos en lecciones previas, mencionamos que el tipo String es una clase de Java, pero con un tratamiento especial. Debido a que este tipo de datos se utiliza en repetidas ocasiones en nuestros programas, es posible omitir el uso de la palabra new, y por tanto simplificar la creación de una variable de tipo String.

Entonces podemos crear un objeto de tipo String de la manera clásica:

```
String nombre = new String("Armando");
```

Pero lo utilizaremos más comúnmente sin el uso de la palabra new, este es el método simplificado:

```
String nombre = "Armando";
```

Ambas opciones son válidas, pero utilizaremos más la segunda opción debido a que se simplifica el código. Sólo no hay que olvidar que el tipo String es una clase y no un tipo primitivo. Este tipo de dato es la única excepción a la regla para crear objetos utilizando el operador new. Todos las demás clases deberán utilizar el operador new para poder crear un objeto.

Empezaremos a utilizar este tipo en nuestros siguientes ejemplos, y así familiarizarnos rápidamente con el uso del tipo String en Java.

EJERCICIOS CURSO FUNDAMENTOS DE JAVA

- **ABRIR LOS ARCHIVOS DE EJERCICIOS EN PDF.**
- **EJERCICIO 15:** Ejercicio Creación de Clase Java.
- **EJERCICIO 16:** Ejercicio Creación de Objetos en Java.

CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

CURSO ONLINE

FUNDAMENTOS DE JAVA

Por: Ing. Ubaldo Acosta



CURSO DE FUNDAMENTOS DE 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.

Además agregamos nuevos cursos para que continúes con tu preparación como programador Java profesional. A continuación te presentamos nuestro listado de cursos:

- | | |
|---|--|
| <ul style="list-style-type: none">✓ Lógica de Programación✓ Fundamentos de Java✓ Programación con Java✓ Java con JDBC✓ HTML, CSS y JavaScript✓ Servlets y JSP's✓ Struts Framework | <ul style="list-style-type: none">✓ 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

Email: informes@globalmentoring.com.mx

