

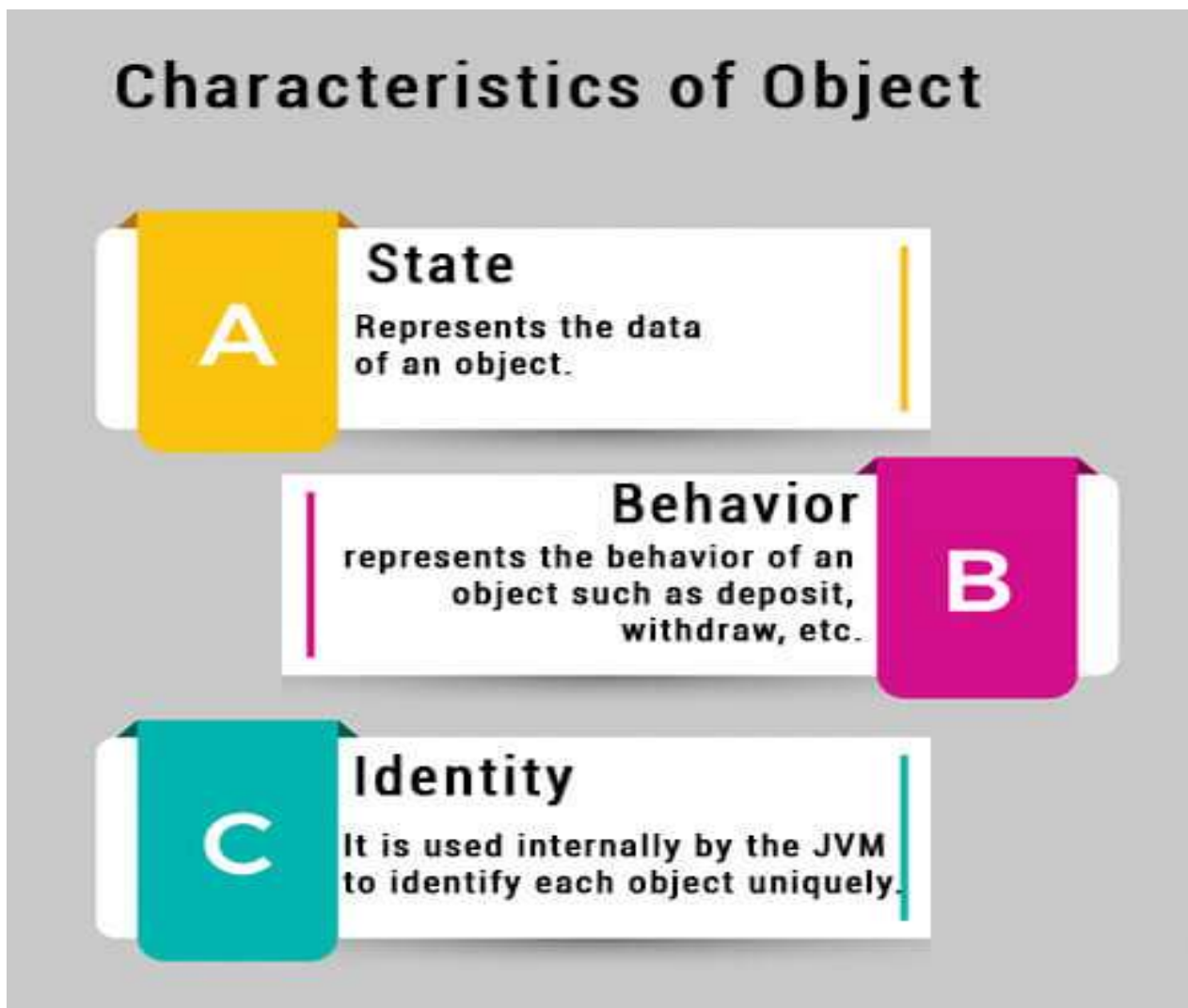
# Objetos y clases en Java

Un objeto en Java es una entidad tanto física como lógica, mientras que una clase en Java es solo una entidad lógica.

## ¿Qué es un objeto en Java?

Es la instancia de una clase donde podemos considerar la clase como el mapa para poder crear el objeto. Desde el momento de la instanciación se convierte en una variable, con su espacio en memoria y lista para ser utilizada. Un objeto tiene tres características:

- **Estado:** representa los datos (valor) de un objeto.
- **Comportamiento:** representa el comportamiento (funcionalidad) de un objeto como depositar, retirar, etc.
- **Identidad:** una identidad de objeto generalmente se implementa a través de una ID única. El valor de la ID no es visible para el usuario externo. Sin embargo, la JVM la utiliza internamente para identificar cada objeto de forma exclusiva.



## ¿Qué es una clase en Java?

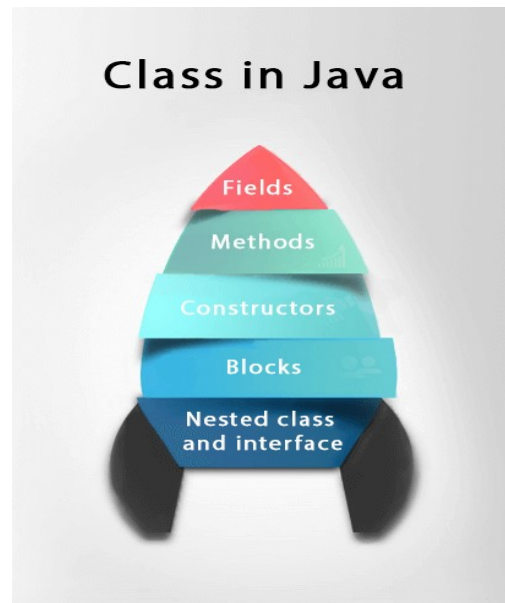
Una clase es un grupo de objetos que tienen propiedades comunes. Es una plantilla o plano a partir del cual se crean los objetos. Es una entidad lógica. No puede ser físico.

Una clase en Java puede contener:

- **Campos**
- **Métodos**
- **Constructores**
- **Bloques**
- **Clase e interfaz anidadas**

Sintaxis para declarar una clase:

```
class <class_name> {  
    campo;  
    método;  
}
```



## Variable de instancia o miembro.

Una variable que se crea dentro de la clase pero fuera del método se conoce como variable de instancia. La variable de instancia no obtiene memoria en tiempo de compilación. Obtiene memoria en tiempo de ejecución cuando se crea un objeto o instancia. Es por eso que se conoce como una variable de instancia.

## Método en Java

En Java, un método es como una función que se utiliza para exponer el comportamiento de un objeto.

Ventaja del método

- Código de reutilización
- Optimización de código

## Palabra reservada new

La nueva palabra clave **new** se utiliza para asignar memoria en tiempo de ejecución. Todos los objetos obtienen memoria en el área de memoria del montón.

Sintaxis para declarar una clase: **TipoDatoObjeto** nombreVariable = **new** TipoDatoObjeto();

## Objeto y clase Ejemplo: principal dentro de la clase

Estamos creando el objeto de la clase Alumno con la palabra clave new e imprimiendo el valor del objeto creando un método main () dentro de la clase.

**Ejemplo:** *A\_EjemploClaseMainEnLaMismaClase.java*

## Objeto y clase Ejemplo: principal fuera de la clase

En el desarrollo en tiempo real, creamos clases y las usamos desde otra clase. Es un enfoque mejor que el anterior. Veamos un ejemplo simple, donde tenemos el método main () en otra clase.

Podemos tener múltiples clases en diferentes archivos Java o un solo archivo Java. Si define varias clases en un solo archivo fuente Java, es una buena idea guardar el nombre del archivo con el nombre de la clase que tiene el método main ().

**Ejemplo:** Archivo: *B\_MainFueraDeLaClase*

## Hay 3 formas de inicializar objeto

Hay 3 formas de inicializar objetos en Java.

1. Por variable de referencia
2. Por método
3. Por constructor

## 1) Objeto y clase Ejemplo: inicialización a través de referencia

Inicializar un objeto significa almacenar datos en el objeto. Veamos un ejemplo simple donde vamos a inicializar el objeto a través de una variable de referencia.

**Ejemplo:** *C\_InicializarObjetoXReferencia.java*

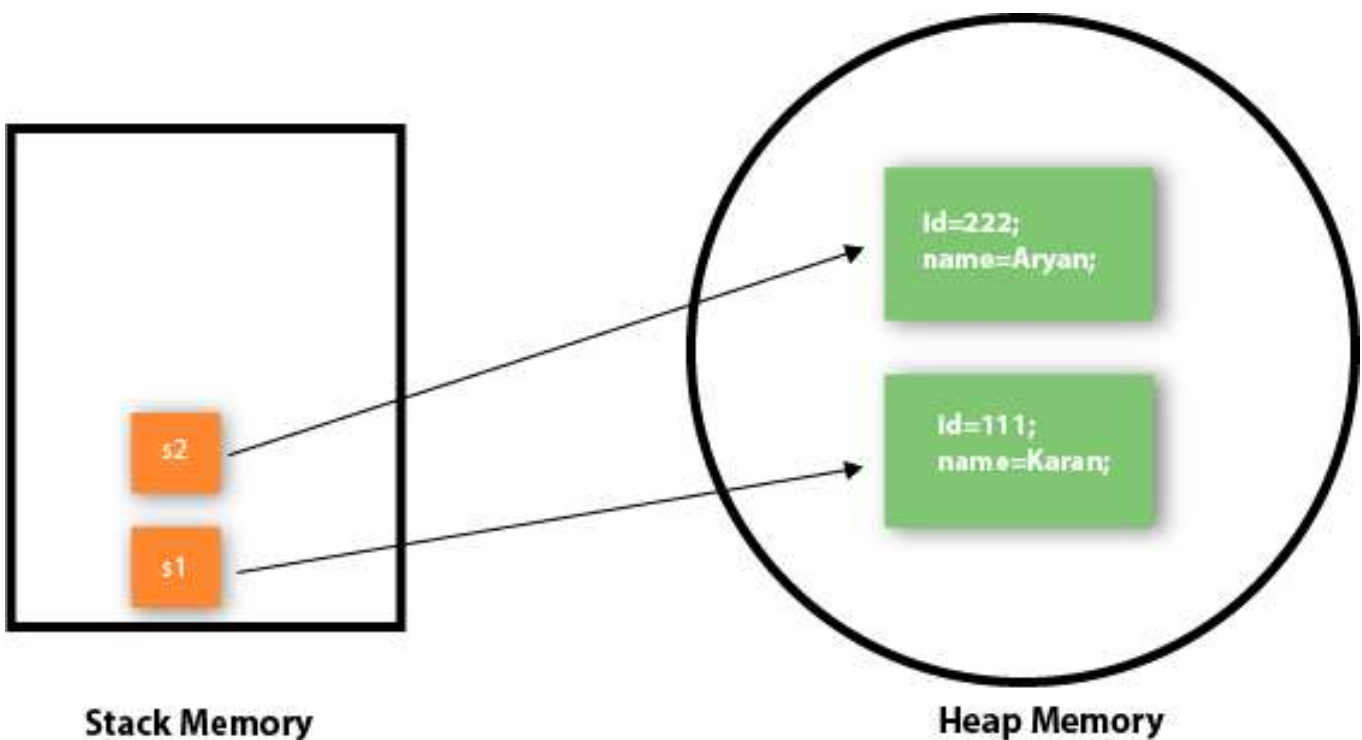
También podemos crear múltiples objetos y almacenar información en ellos a través de la variable de referencia.

**Ejemplo:** *D\_InicializarMultiplesObjetosxReferencia.java*

## 2) Objeto y clase Ejemplo: inicialización a través del método

En este ejemplo, estamos creando dos objetos de la clase Estudiante e inicializando el valor de estos objetos invocando el método insertar(). Aquí, estamos mostrando el estado (datos) de los objetos invocando el método mostrar ().

**Archivo:** *E\_InicializacionPorMetodo.java*



Como puede ver en la figura anterior, el objeto obtiene la memoria en el área de memoria de almacenamiento dinámico. La variable de referencia se refiere al objeto asignado en el área de memoria de almacenamiento dinámico. Aquí, s1 y s2 son variables de referencia que se refieren a los objetos asignados en la memoria.

## 3) Ejemplo de objeto y clase: inicialización a través de un constructor

Aprenderemos sobre constructores detalladamente mas adelante.

## Objeto y clase:

Veamos un ejemplo donde mantenemos registros de empleados.

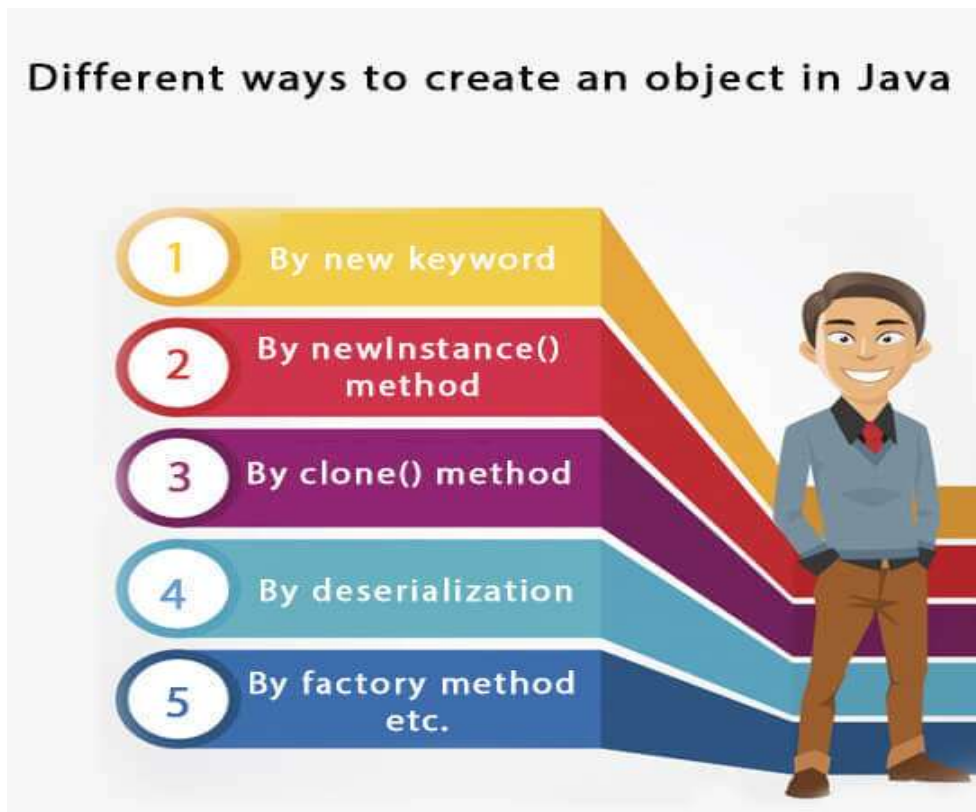
**Archivo:** *F\_InicializacionPorConstructor.java*

**Archivo:** *G\_MultiplesObjetosMismoTipo.java*

## ¿Cuáles son las diferentes formas de crear un objeto en Java?

Hay muchas formas de crear un objeto en java. Son:

- Por nueva palabra clave
- Por el método newInstance ()
- Por el método clone ()
- Por deserialización
- Usando el patrón de diseño fabric.



## Objeto anónimo

Anónimo simplemente significa sin nombre. Un objeto que no tiene referencia se conoce como un objeto anónimo. Se puede usar solo en el momento de la creación de objetos.

Si tiene que usar un objeto solo una vez, un objeto anónimo es un buen enfoque. Por ejemplo:

**Archivo:** *H\_ObjetoAnonimo.java*

## Crear múltiples objetos por un solo tipo

Podemos crear múltiples objetos por un solo tipo como lo hacemos en caso de primitivas.

**Archivo:** *H\_ObjetoAnonimo.java*

**Archivo:** *J\_EjemploCuentasBancarias*