



El futuro digital  
es de todos

MinTIC



Vigilada Mineducación

## CICLO II: Programación Básica en Java

Misión  
TIC2022





El futuro digital  
es de todos

MinTIC



Vigilada Mineducación

# Sesión 6: Introducción a Java

Programación Orientada a Objetos (POO)





# Objetivos de la sesión

Al finalizar esta sesión estarás en capacidad de:

1. Especificar cómo se inicializa un objeto cuando no existe un constructor
2. Inicializar objetos utilizando un constructor
3. Definir el diagrama de clase insertando el constructor
4. Mostrar la existencia de los paquetes del API de Java y explicar los más importantes.
5. Explicar los conceptos asociados a métodos y campos static



# ¿Qué es un constructor en Java?

- En Java es un método especial dentro de una clase, que se llama automáticamente cada vez que se crea un objeto de esa clase.
- Un constructor inicializa un objeto cuando se crea. Tiene el mismo nombre que su clase y es sintácticamente similar a un método. Sin embargo, los constructores no tienen un tipo de devolución explícito.
- Todas las clases tienen constructores, ya sea que se defina uno o no, porque Java proporciona automáticamente un constructor predeterminado. En este caso, las variables de miembro no inicializadas tienen sus valores predeterminados, que son cero, null y false. Una vez que defines tu propio constructor, el constructor predeterminado ya no se usa.



# Constructor – Sintaxis en Java

```
class className {  
  
    // Constructor  
    nombre de la clase() {  
        // ...  
    }  
  
    // Otros metodos...  
    void method1 () {  
        // ...  
    }  
  
    void method2 () {  
        // ...  
    }  
}
```



# Ejemplo – Estructura en Java

El constructor de una clase tiene las siguientes características:

- Tiene el mismo nombre que la clase a la que pertenece.
- En una clase puede haber varios constructores con el mismo nombre y distinto número de argumentos (se puede sobrecargar)
- No se hereda.
- No puede devolver ningún valor (incluyendo void).
- Debe declararse público (salvo casos excepcionales) para que pueda ser invocado desde cualquier parte donde se desee crear un objeto de su clase.

```
class Vehiculo {  
    int pasajeros; //números de pasajeros  
    int capacidad; //capacidad del combustible en galones  
    int mpg;       //combustible consumido en miles por galon  
  
    //Esto es un constructor para Vehiculo  
    Vehiculo (int p, int c, int m) {  
        pasajeros=p;  
        capacidad=c;  
        mpg=m;  
    }  
}
```



# Ejemplo simple de un constructor:

```
public class PruebaPrecio {  
    public static void main(String [] args) {  
        Precio p;           // Crea una referencia de la clase Precio  
        p = new Precio();    // Crea el objeto de la clase Precio y realiza  
                             // una llamada al metodo constructor  
    }  
}
```

En el ejemplo de la clase PruebaPrecio, que utiliza una instancia de la clase Precio, la llamada al constructor se produce en la sentencia `p = new Precio();`. Mientras que la ejecución de `new` genera una nueva instancia y devuelve su dirección de memoria, la ejecución del constructor `Precio()` inicializa los valores de los atributos. Para este caso se inicializa un objeto cuando “no existe” un constructor.



# Ejemplo constructor instanciación de la clase anterior

```
class MiClase {  
    int x;  
    MiClase() {  
        x=10;  
    }  
}  
  
class ConstructorDemo {  
    public static void main(String[]  
args) {  
        MiClase t1= new MiClase();  
        MiClase t2= new MiClase();  
        System.out.println(t1.x + " -  
"+t2.x);  
    }  
}
```

En este ejemplo, el constructor de MiClase es:

```
MiClase(){  
    x=10;  
}
```

Este constructor asigna a la variable de instancia x de MiClase el valor de 10. Este constructor es llamado por new cuando se crea un objeto.

Salida: 10 - 10



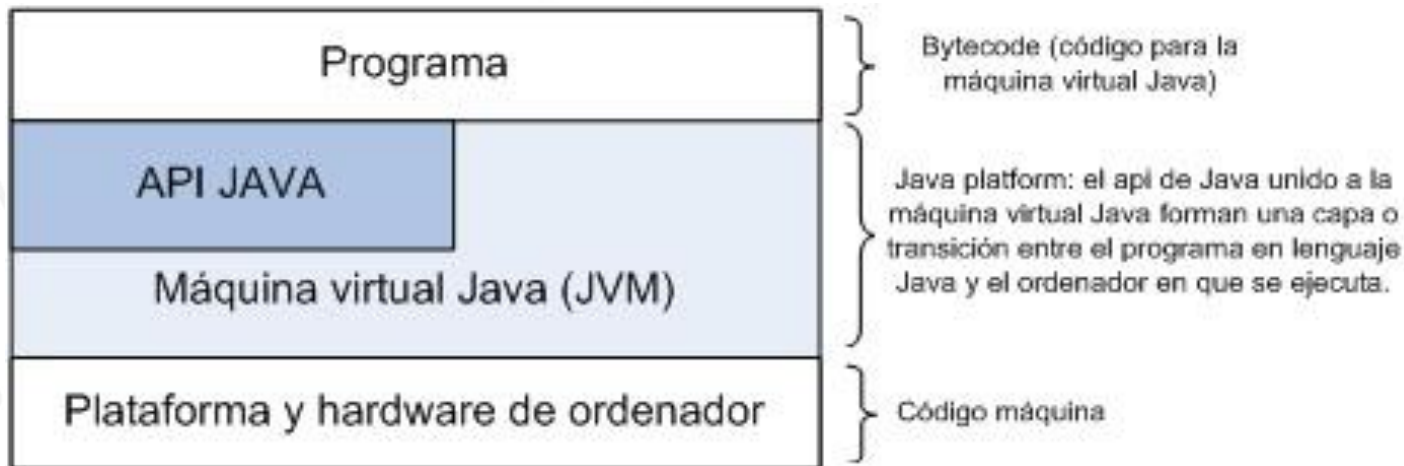


# API de Java

- Los programas se construyen combinando nuevos métodos y clases con los predefinidos en la API de Java.
- Las clases son agrupadas en paquetes y pueden ser importadas y reusadas.
- La API de Java contiene una colección enorme de clases predefinidas que contienen métodos para ejecutar cálculos matemáticos, manipulación de strings, de caracteres, operaciones de entrada/salida, operaciones de bases de datos, archivos, y muchas más.



# API de Java





# Métodos static y atributos static

La mayoría de los métodos se ejecutan en respuesta a las llamadas que se hacen de ellos en objetos específicos, sin embargo, hay otros que no. En ocasiones un método realiza una tarea que no depende del contenido de ningún objeto. En este caso se aplica a la clase en la que está declarado como un todo y se conoce como método static.

Consideraciones:

- Sólo existen una vez por clase, independientemente del número de instancias (objetos) de la clase que se haya creado y aunque no exista ninguna.
- El método o el atributo se comportan siempre de la misma manera
- Se puede acceder a los miembros estáticos utilizando el nombre de la clase.
- Un método estático no puede acceder a miembros no estáticos directamente, tiene que crear primero un objeto



# Métodos static y atributos static

## Algunas Reglas:

- Los métodos y atributos estáticos se llaman así:  
NombreClase.metodoEstatico(<args>;  
NombreClase.atributoEstatico;
- Para acceder a los métodos y atributos **no** estáticos se necesita de una instancia (objeto) de la clase:  
NombreClase nombreObjeto = new NombreClase();
- Los miembros **no** estáticos se llaman así:  
nombreObjeto.metodoNormal();  
nombreObjeto.atributoNormal;



# Métodos static y atributos static

Algunas Reglas:

- Cuando la llamada a un método o atributo estático de la clase se realiza dentro de la propia clase se puede omitir el nombre de la misma:

```
metodoEstatico();  
atributoEstatico;
```

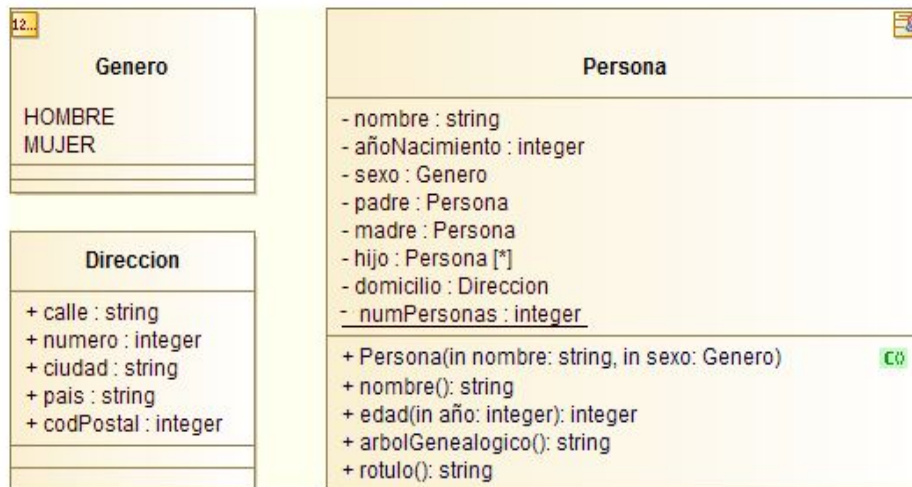
en vez de:

```
NombreClase.metodoEstatico();  
NombreClase.atributoEstatico;
```



# Atributos estáticos

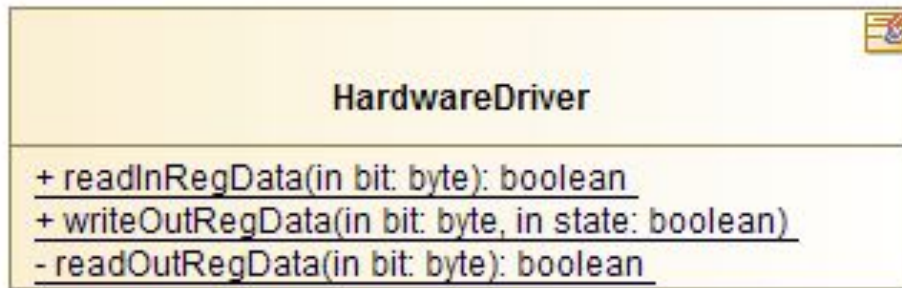
En la representación de Diagrama de clase, los atributos estáticos se representan subrayados, los públicos con un prefijo + y los privados con un prefijo -.





# Métodos estáticos

En la representación de Diagrama de clase, los métodos estáticos se representan subrayados.







El futuro digital  
es de todos

MinTIC

**UN** UNIVERSIDAD  
**DEL NORTE**

Vigilada Mineducación

**¡GRACIAS**  
**POR SER PARTE DE**  
**ESTA EXPERIENCIA**  
**DE APRENDIZAJE!**



**Misión**  
**TIC 2022**