

Conceptos de POO en JAVA

Índice

1. [Atributos](#)
2. [Modificadores de acceso](#)
3. [Constructores](#)
4. [Métodos](#)
5. [Setters y Getters](#)



01 Atributos

Los atributos de la clase Artículo

Necesitamos crear una clase **Artículo**, la información con la que contamos es:

- Un Artículo tiene una descripción
- Tiene un precio de venta
- Tiene un stock

Esta clase debe poder responder si hay stock y cuál es su precio.

Diseño de la clase Artículo

Artículo

- descripción :String
- precioVenta: double
- stock: int

- + Artículo(descripcion:String, cantidad:int, precio:double)
- + boolean hayStock()
- + double consultarPrecio()

02

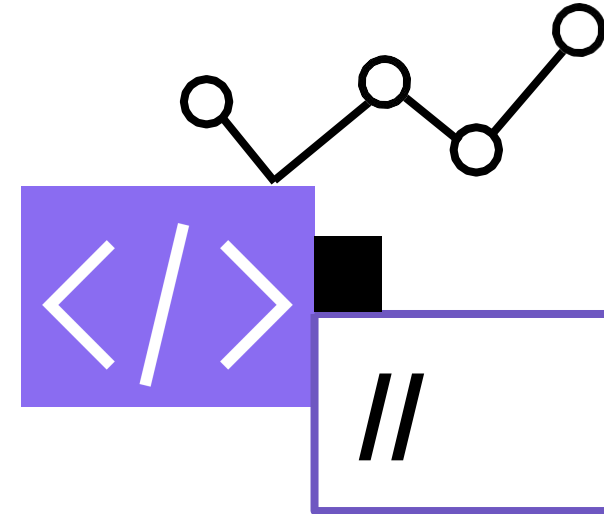
Modificadores de acceso

Accediendo a nuestra clase Artículo

Para definir cuándo un atributo, método o constructor puede ser accesible o no, usamos las palabras reservadas: **Public, Private o Protected**

Entonces, al usar este concepto aplicado a nuestra clase artículo **es necesario que sus atributos tengan un modificador de acceso de tipo Private** ya que nos interesa que el código solo sea accesible dentro de la clase declarada.

Si necesitamos acceder a ellos para ver o cambiar tales atributos necesitamos agregar métodos de acceso setters y getters que veremos con detalle mas adelante.



Los atributos de la clase Articulo en Java

{}

```
public class Articulo{  
  
    private String descripcion;  
    private double precioVenta;  
    private int stock;  
  
}
```

03 Constructores

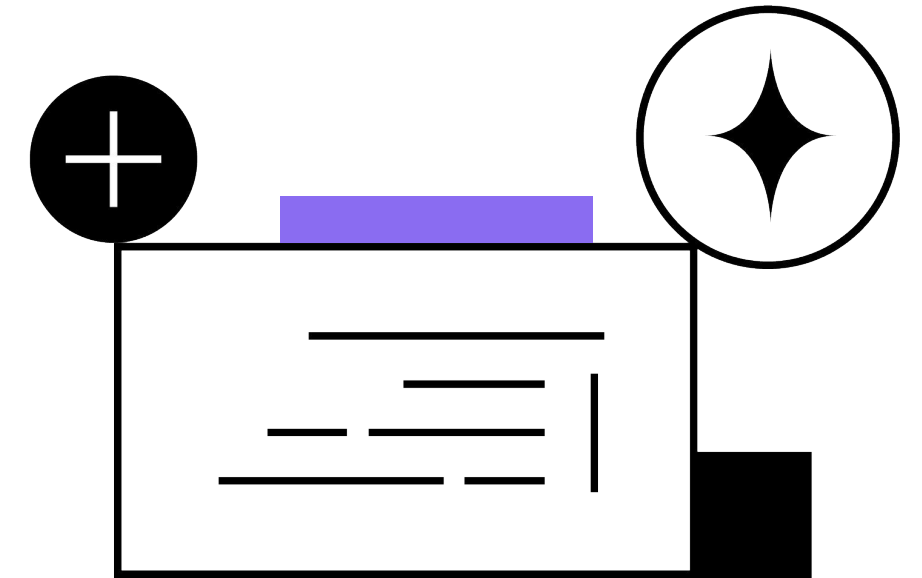
Constructores en la clase Artículo

Artículo

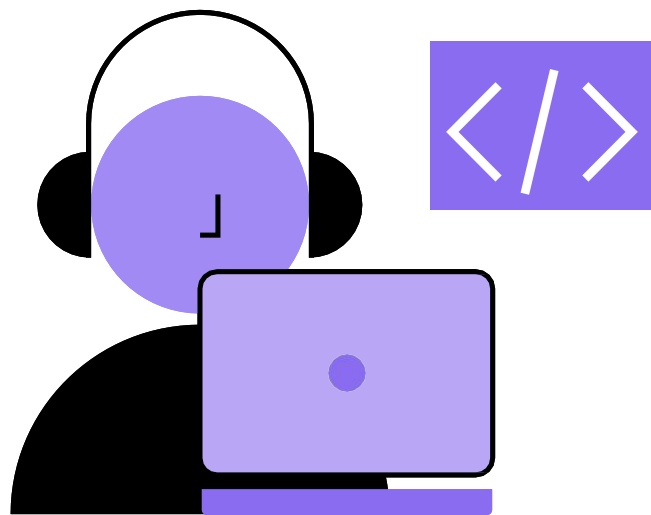
- descripción :String
- precioVenta: double
- stock: int

+ Artículo(descripcion:String,
cantidad:int, precio:double)

- + boolean hayStock()
- + double consultarPrecio()

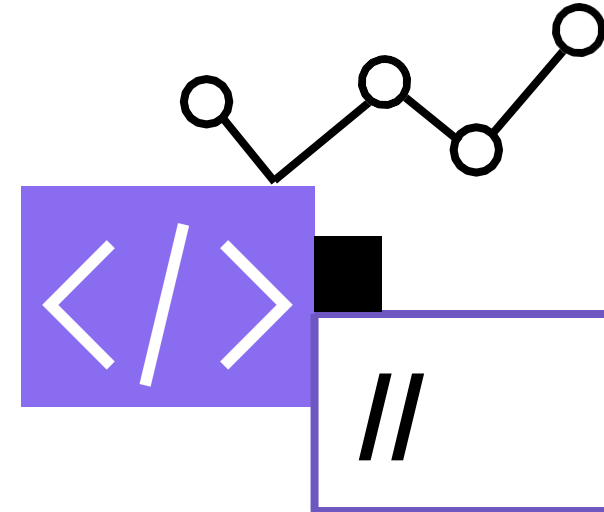


Los métodos de la clase Artículo en Java



```
public class Artículo{
    private String descripcion;
    private double precioVenta;
    private int stock;

    public Artículo(String descripcion, int cantidad, double
precio){
        this.descripcion=descripcion;
        precioVenta=precio;
        stock=cantidad;
    }
}
```



{código
}

```
public class Artículo{  
    private String descripcion;  
    private double precioVenta;  
    private int stock;
```

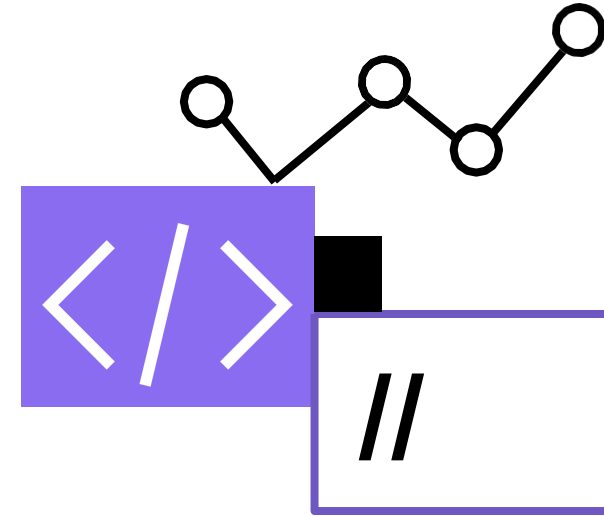
```
    public Artículo(String descripcion,  
                      int cantidad, double precio){  
        this.descripcion=descripcion;  
        precioVenta= precio;  
        stock=cantidad;  
    }
```

El constructor es un método que no tiene tipo de dato, se llama igual que la clase. Recibe como parámetros los valores que se desea asignar inicialmente a los atributos, es decir, los valores iniciales. Se puede usar para inicializar los atributos.

{código }

```
public class Artículo{  
    private String descripcion;  
    private double precioVenta;  
    private int stock;  
  
    public Artículo(String descripcion,  
int cantidad,double precio){  
        this.descripcion=descripcion;  
        precioVenta= precio;  
        stock=cantidad;  
    }  
}
```

Para diferenciar el atributo descripción del parámetro que tiene el mismo nombre, usamos **this**.
This hace referencia al objeto o instancia con la que se está trabajando.



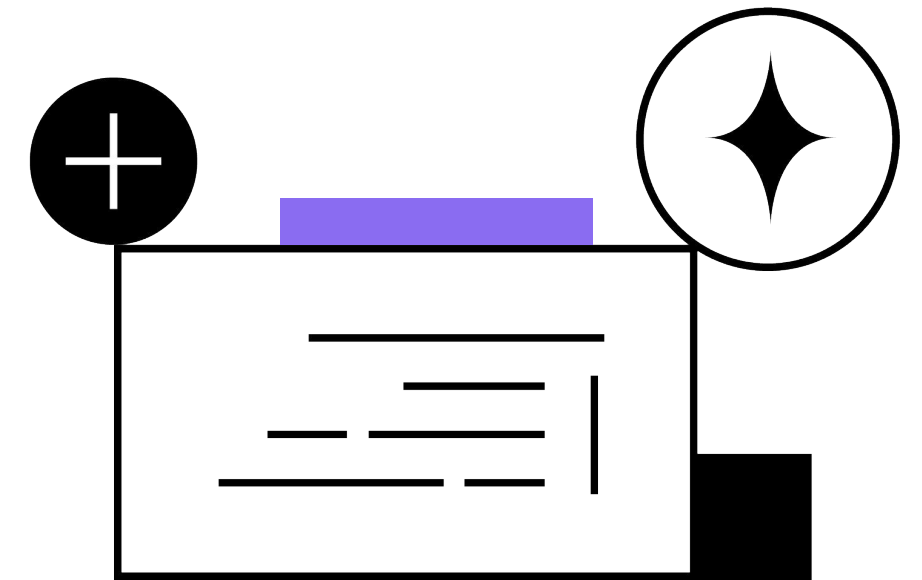
04 Métodos

Los métodos de la clase Artículo

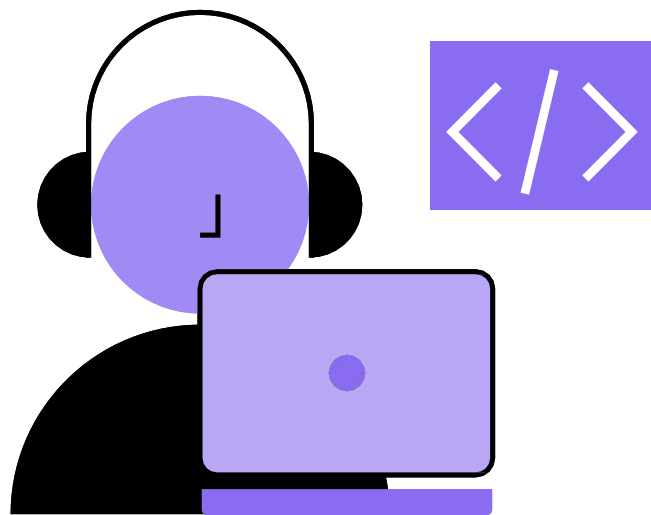
Artículo

- descripción :String
- precioVenta: double
- stock: int
- + Artículo(descripcion:String, cantidad:int, precio:double)
- + boolean hayStock()
- + double consultarPrecio()

Los métodos de nuestra clase son hayStock() y consultarPrecio()



Los métodos de la clase Artículo en Java



```
public class Artículo{
    private String descripcion;
    private double precioVenta;
    private int stock;

+ public Artículo(String descripcion, int cantidad, double precio)

    public boolean hayStock(){
        return stock>0;
    }
    public double consultarPrecio(){
        return precioVenta;
    }
}
```

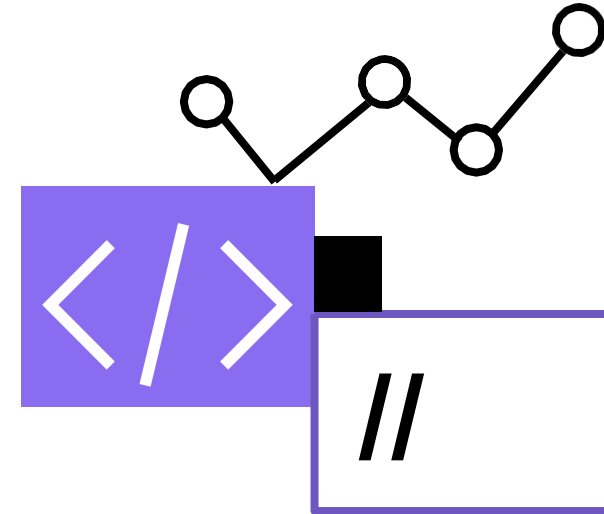
{código
}

```
public class Artículo
    private String descripcion;
    private double precioVenta;
    private int stock;
+ public Artículo(String    descripcion,    int
cantidad,double precio)
```

```
public boolean hayStock(){
    return stock>0;}
```

```
public double consultarPrecio()
    return precioVenta;
```

Método hayStock()
Devuelve **true** si stock
es mayor a 0 y **false** en
caso contrario

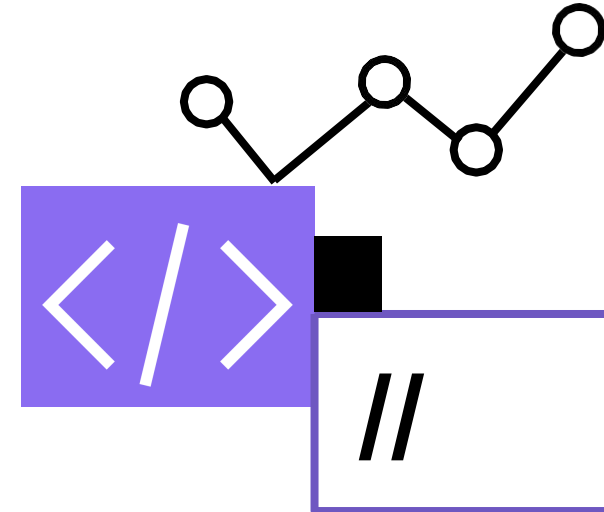


{código
}

```
public class Artículo
    private String descripcion;
    private double precioVenta;
    private double precioCompra;
    private int stock;
+ public Artículo(String descripcion, int
cantidad, double precio)
    public boolean hayStock()
        return stock > 0;

    public double consultarPrecio(){
        return precioVenta;
    }
```

Método consultarPrecio()
Devuelve el **Precio de
Venta**



05 Setters y Getters

Setters y Getters

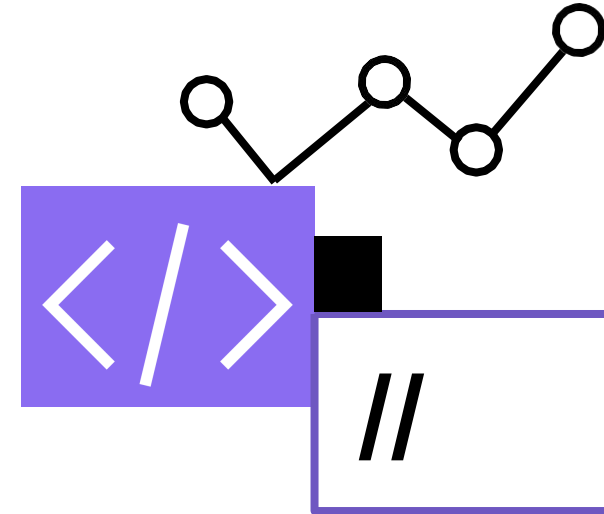
Para lograr y proteger el encapsulamiento de nuestros atributos en la clase articulo debemos definirlos como **privados** y agregar getters y setters.

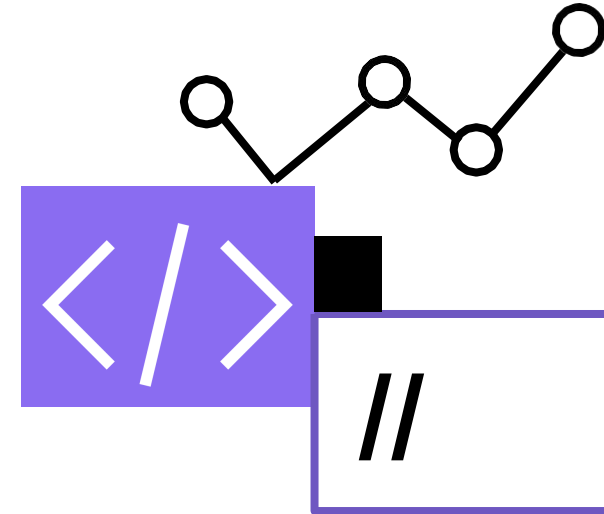
Recuerden que es una buena práctica impedir que cualquier otro objeto pueda tener acceso a la estructura interna de un objeto, es decir, preservar el **encapsulamiento**.

{ código }

```
public String getDescripcion(){  
    return descripcion;  
}  
public double getPrecioVenta(){  
    return precioVenta;  
}  
public int getStock(){  
    return stock;  
}  
  
public void setDescripcion(String  
descripcion){  
    this.descripcion= descripcion;  
}  
public void setPrcioVenta(double precio){  
    precioVenta=precio;  
}  
public void setStock(int stock){  
    this.stock=stock;  
}
```

Los métodos **get** permiten acceder al valor de un atributo para una consulta o para usar ese valor en otra operación.

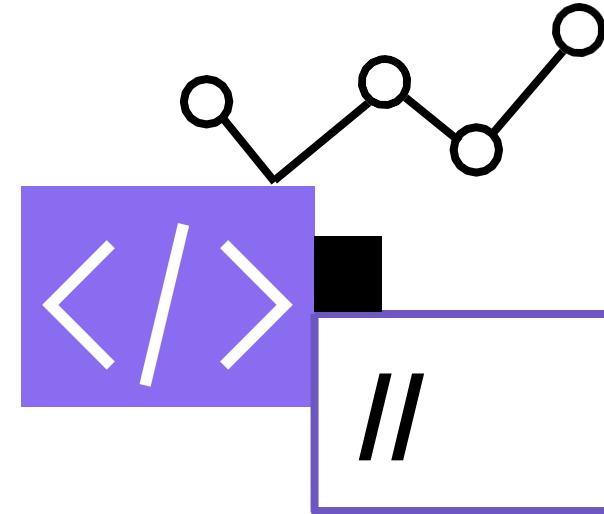




{código
}

```
public String getDescripcion(){  
    return descripcion;  
}  
public double getPrecioVenta(){  
    return precioVenta;  
}  
public int getStock(){  
    return stock;  
}  
public void setDescripcion(String descripcion){  
    this.descripcion= descripcion;  
}  
public void setPrcioVenta(double precio){  
    precioVenta=precio;  
}  
public void setStock(int stock){  
    this.stock=stock;  
}
```

Los métodos **set** permiten cambiar el valor de un atributo, reciben por parámetro el nuevo valor y lo asignan al atributo correspondiente.



{código }

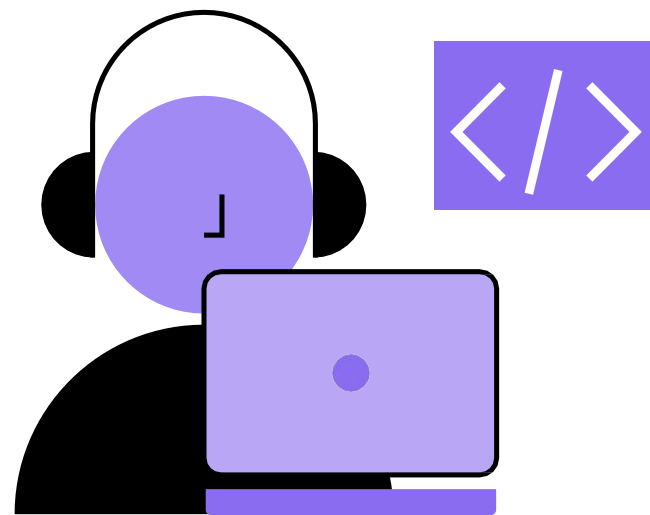
```
public class Artículo{  
    private String descripcion;  
    private double precioVenta;  
    private int stock;  
  
    public Artículo(String descripcion, int  
cantidad,double precio){  
        this.descripcion=descripcion;  
        precioVenta= precio;  
        stock=cantidad;  
    }  
}
```

Para diferenciar el atributo **descripción** del parámetro que tiene el mismo nombre, usamos **this**. This hace referencia al objeto o instancia con la que se está trabajando.

No olviden que:

- Cuando definamos un objeto, debemos dejar sus atributos privados.
- Los métodos que sean públicos serán vistos por los otros objetos.
- Usar siempre métodos públicos para ver o modificar las características de los objetos.
- Para cambiar el valor de un atributo se usa un método set, por ejemplo, para cambiar el nombre será `setNombre(String)`
- Para obtener el valor de un atributo se usa un método get, por ejemplo, para saber el nombre será `getNombre(): String`

Los atributos, métodos y constructores de nuestra clase Artículo se verían así:



```
public class Artículo{
    private String descripcion;
    private double precioVenta;
    private int stock;

    public Artículo(String descripcion, int cantidad, double
    {precio)
        this.descripcion=descripcion;
        precioVenta=precio;
        stock=cantidad;
    }

    public String getDescripcion(){
        return descripcion;}
    public double getPrecioVenta(){
        return precioVenta;}
    public int getStock(){
        return stock;}
    public void setDescripcion(String descripcion)
    {
        this.descripcion= descripcion;}
    public void setPrcioVenta(double precio){
        precioVenta=precio; }
    public void setStock(int stock){
```


¡Muchas
gracias!