



UNJu | Universidad
Nacional de Jujuy



Argentina
programa
4.0

Constructores y Garbage Collector





Java | Constructor

Los constructores son un tipo específico de método pertenecientes a la clase. Se utilizan para construir o instanciar objetos de una clase. Su nombre es idéntico al de la clase en la que se ha definido y no tiene ningún tipo de retorno. Puede haber varios constructores de acuerdo a la necesidad del usuario.

Sintaxis

```
<modificador de acceso> class  
nombreClase([lista_de_parametros]) {  
    [declaracion_de_variables_locales]  
    Sentencias;  
}
```





Java | Tipos de constructores

- Constructor por defecto
- Constructor de copia
- Constructor personalizado

```
public class Persona {  
    private int enteroPrivado;  
  
    public Persona() { // constructor por defecto  
        enteroPrivado=0;  
    }  
  
    public Persona(int entero) { // constructor con parámetros  
        enteroPrivado = entero;  
    }  
}
```





Java | Constructor por defecto

Si no se define ningún constructor de una clase, el compilador generará un constructor por defecto. El constructor por defecto no tiene argumentos y simplemente sitúa ceros en cada byte de las variables de instancia de un objeto. Si se definen constructores para una clase, el constructor por defecto no se genera.

```
public Persona() { // constructor por defecto  
    enteroPrivado=0;  
}
```





Java | Constructor de copia

El constructor de copia se utiliza para inicializar un objeto con otro objeto de la misma clase.

Construye un nuevo objeto como una copia del que se le pasa.

```
public Cuenta (Cuenta otra) {  
    codigo = otra.codigo;  
    saldo = otra.saldo;  
    titular = otra.titular;  
}
```



Java | Constructor personalizado



El constructor común (personalizado) es aquel que recibe parámetros para asignarles como valores iniciales a los datos miembros, o sea que al crear la instancia, se pasó unos parámetros para inicializar .

Si se declara algún constructor, entonces ya no se puede usar el constructor personalizado. Es necesario usar el constructor declarado en la clase.

- ❖ De entre todos los constructores que existan, tan sólo uno se ejecutará al crear un objeto de la clase.



Java | Destructor



El destructor se utiliza para destruir una instancia de una clase y liberar memoria. Su nombre es el mismo que el de la clase, precedido por el carácter ~
No puede recibir ningún parámetro y, como el constructor, no tiene ningún tipo de retorno.

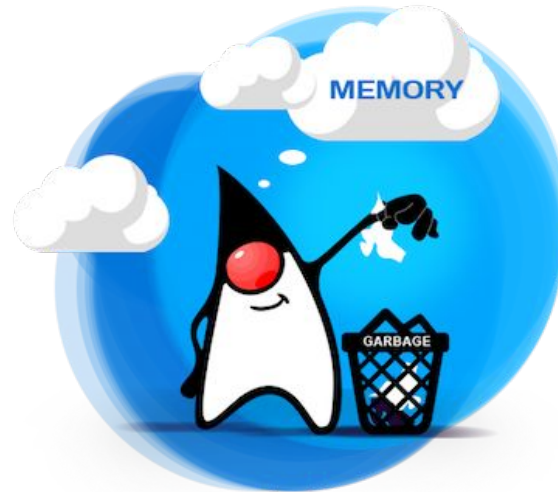


Java | Garbage Collector



En Java no hay destructores, ya que la liberación de memoria es llevada a cabo por el Garbage Collector cuando las instancias de los objetos quedan desreferenciadas.

El método `dispose()` de cada objeto se llama previo a ser “recolectado”.





UNJu | Universidad
Nacional de Jujuy



Argentina
programa
4.0

RELACIÓN ENTRE CLASES Y OBJETOS



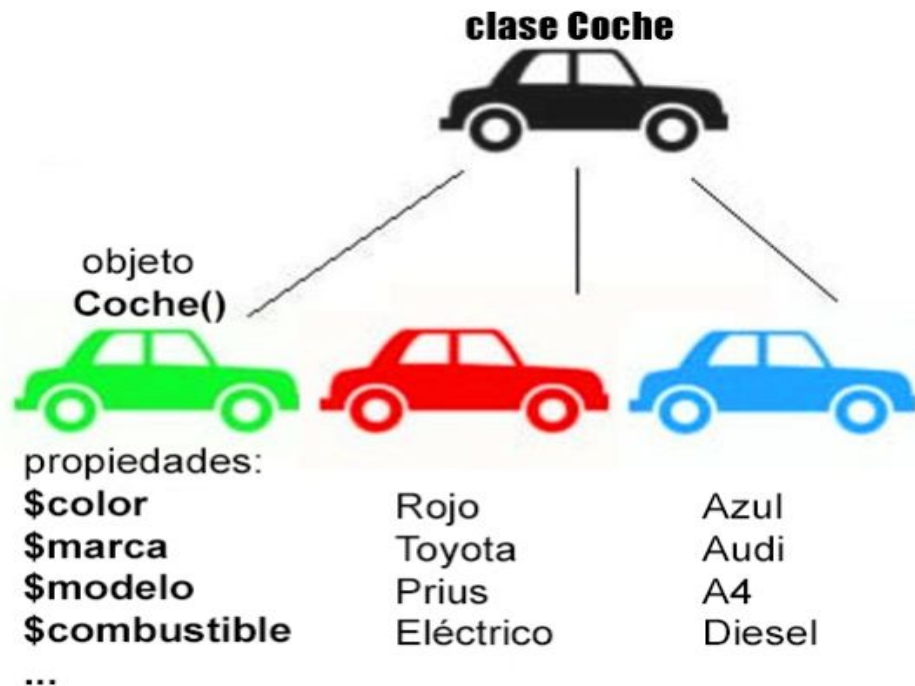
Java | Relación entre objeto y clase



Un objeto es una INSTANCIA de una clase. Por lo tanto, los objetos hacen uso de los Atributos (variables) y Métodos (Funciones y Procedimientos) de su correspondiente Clase.



Java | Relación entre objeto y clase



MÉTODOS

cargarCombustible()
arrancarMotor()
frenar()
acelerar()
....

Java | Relación entre objeto y clase



```
public class Coche{  
    private String color;  
  
    private String marca;  
  
    private String modelo;  
  
    private String combustible;  
  
    ....  
  
    public void setMarca(String marca) {  
        this.marca = marca;  
    }  
  
    .....
```

```
public class Main {  
  
    public static void main(String[] args) {  
        Coche miCoche = new Coche();  
  
        miCoche.setMarca("Toyota" );  
  
        String cocheMarca = "La marca de mi auto es: " +  
            miCoche.marca;  
  
        System.out.println(cocheMarca);  
    }  
}
```





UNJu | Universidad
Nacional de Jujuy



Argentina
programa
4.0

FUNDAMENTOS DE POO



Java | Fundamentos de POO



POO se ha convertido durante las pasadas dos décadas en el paradigma de programación dominante, y en una herramienta para resolver la llamada crisis del software.

“El paradigma orientado a objetos es una metodología de desarrollo de aplicaciones en la cual éstas se organizan como colecciones cooperativas de objetos, cada uno de los cuales representan una instancia de alguna clase, y cuyas clases son miembros de jerarquías de clases unidas mediante relaciones de herencia” (Grady Booch)



Java | Fundamentos de POO



Ahora analicemos el mecanismo que utilizó Luis para resolver el problema:

1. Encontrar un agente apropiado (Pedro)
2. Enviarle un mensaje conteniendo la petición (enviar flores a María)
3. Es la responsabilidad de Pedro cumplir esa petición
4. Para ello, es posible que Pedro de algún modo disponga de algún método (algoritmo o conjunto de operaciones) para realizar la tarea
5. Luis no necesita (ni le interesa) conocer el método particular que Pedro utilizará para la petición (esa información es oculta)





Java | Fundamentos de POO

Repasemos algunos conceptos importantes:

- Agentes y comunidades: Un programa OO se estructura como una comunidad de agentes que interaccionan (OBJETOS).
- Mensaje y métodos: A un objeto se le envían mensajes para que realice una determinada acción. El objeto selecciona un método apropiado para realizar dicha acción y a este proceso se le denomina Paso de mensajes.
- Responsabilidades: El comportamiento de cada objeto se describe en términos de responsabilidades (hay mayor independencia entre los objetos).





Java | Fundamentos de POO

-Objetos y clases: Un objeto es una encapsulación de un estado (valores de los datos) y comportamiento (operaciones). Otra manera de pensar en un objeto es como la instancia de una clase.

-Jerarquías de clases: Las clases de objetos pueden ser organizadas en una estructura jerárquica de herencia. Una clase 'hijo' hereda propiedades de una clase 'padre' más alta en la jerarquía (más general)

-Enlace de métodos: Instante en el cual una llamada a un método es asociada al código que se debe ejecutar. Este enlace puede ser estático (en tiempo de compilación), y dinámico (en tiempo de ejecución)



Java | Fundamentos de POO



Características generales de POO, podemos mencionar algunos más de manera resumida:



Java | Fundamentos de POO



En definitiva, el objeto de POO es producir aplicaciones + fáciles de cambiar (MANTENIBILIDAD)

El Enfoque Orientado a Objetos se basa en cuatro principios que constituyen la base de todo desarrollo orientado a objetos. Estos son:

- Abstracción
- Encapsulamiento
- Modularidad
- Herencia

Y otros elementos a destacar (aunque no fundamentales):

- Polimorfismo
- Tipificación
- Concurrencia
- Persistencia





UNJu | Universidad
Nacional de Jujuy

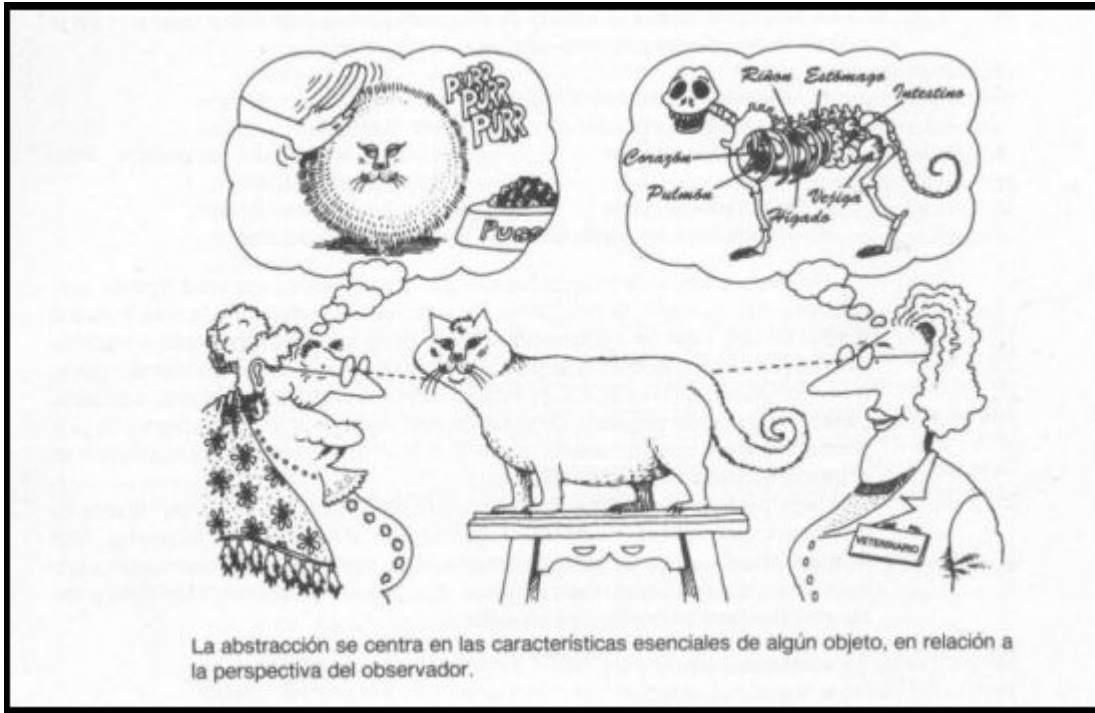


Argentina
programa
4.0

ABSTRACCIÓN



Java | Abstraccion





UNJu | Universidad
Nacional de Jujuy

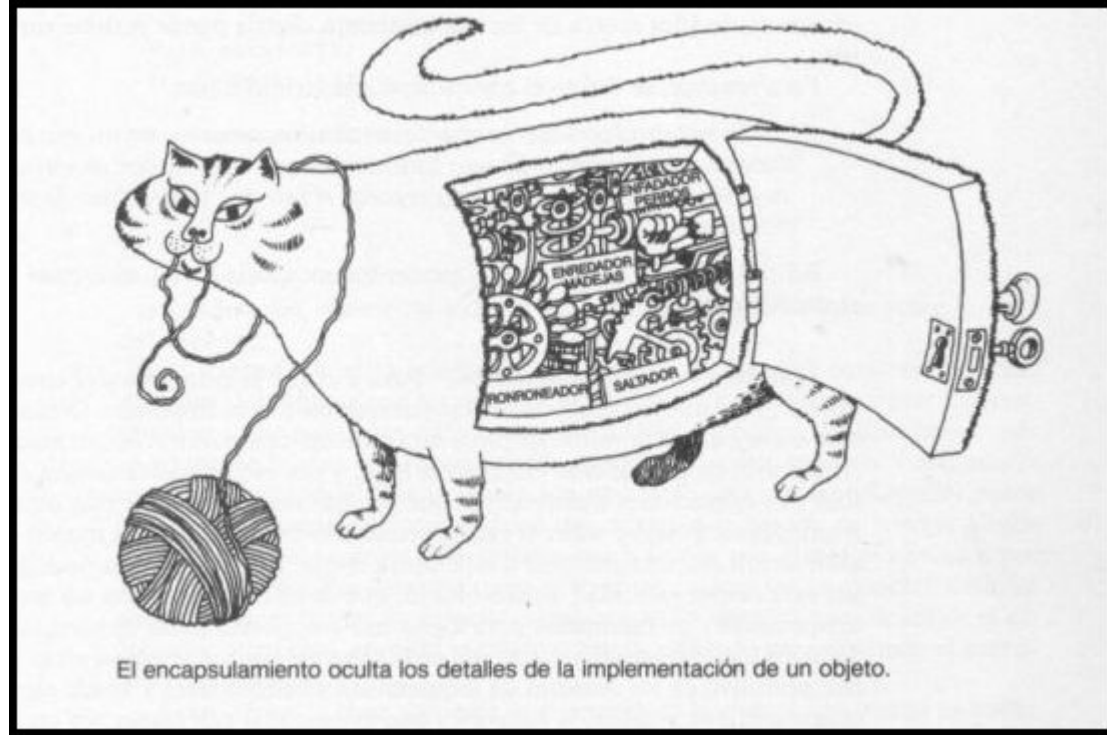


Argentina
programa
4.0

ENCAPSULAMIENTO



Java | Encapsulamiento



Java | Abstracción vs Encapsulamiento



1. Abstracción: Busca la solución en el diseño. Únicamente información relevante, Centrado en la ejecución.
2. Encapsulamiento: Busca la solución en la implementación. Ocultación de código para protegerlo. Centrado en la ejecución.

