



Programación Orientada a Objetos

Autor(es):

Gustavo Eduardo Gil Prado
Docente TC

Resumen:

El objetivo de esta presentación presentar las bases conceptuales, como herramienta en el curso de Programación Orientada a Objetos

FACULTAD DE INGENIERÍA

Ingeniería Informática
Institución Universitaria Colegio Mayor del Cauca



Guía de Curso – 30/05/2022

Descripción:

Esta presentación tienen la intención de servir de apoyo al desarrollo del curso de Programación Orientada a Objetos

Palabras clave:

Algoritmo, Clase, Instancia, Aplicación, Atributo, Método

Referencie este documento así: Gil, G.E. (2023). Programación Orientada a Objetos [Guía de Curso]. Institución Universitaria Colegio Mayor del Cauca.

Programación Orientada a Objetos

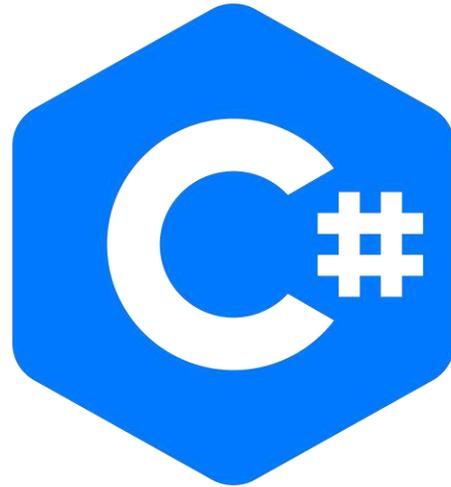
Docente: Gustavo Eduardo Gil Prado

Introducción



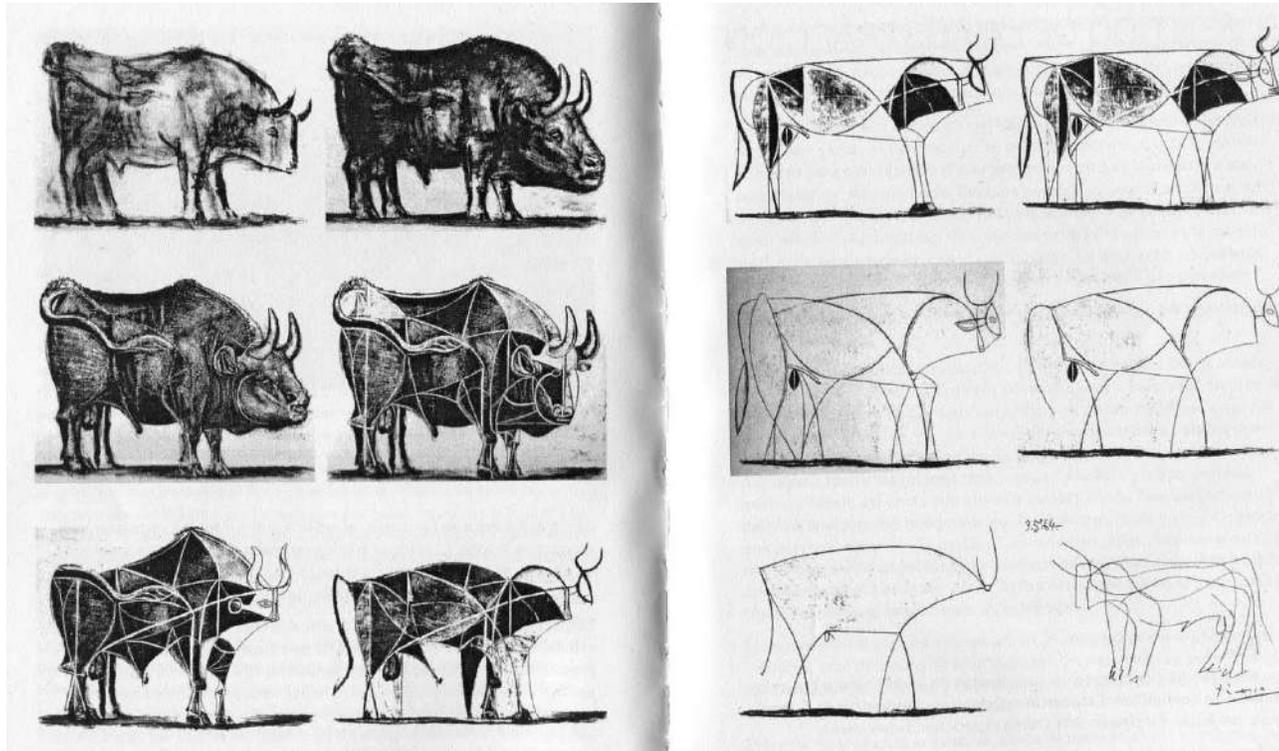
Principales características

- Abstracción
- Encapsulamiento
- Polimorfismo
- Herencia
- Modularidad
- Principio de ocultación
- ...



[Esta foto](#) de Autor desconocido está bajo licencia [CC BY-NC-ND](#)

Abstracción



Esta foto de Autor desconocido está bajo licencia [CC BY-NC-ND](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Abstracción

Supresión intencionada de algunos de las características y o detalles de un proceso o artefacto, lo que permite destacar algunas características, haciendo mas fácil la solución de problemas.

Es una forma habitual para trabajar en entornos o con problemas complejos

La abstracción permite crear un modelo a partir de la realidad

Es una forma de trabajo aplicada a múltiples campos.

Abstracción

Mecanismos de abstracción en programación además se asocian con otras técnicas como lo son:

Ocultamiento de la información: Omisión de partes de la implementación, mediante una interfaz que simplifique su interacción.

Programación modular: Método con el que se abordan problemas de cualquier tamaño, mediante la premisa de descomponer la solución en módulos independientes de programación.

Programación modular y la abstracción...

Continuación

La descomposición en módulos debe tener las siguientes características para asegurar una solución de calidad:

Conexiones mínimas entre el los componentes del programa y módulos.

Los módulos deben realizar tareas definidas manteniendo independencia del programa principal.

Las soluciones deben ser generales, permitiendo ser utilizadas por otros programas.

Programación modular y la abstracción...

Continuación

Beneficios:

Facilidad para la solución de problemas mediante software

Mantenimiento del código

- Facilidad en agregar nuevas características
- Pruebas y depuración mas simples.

Incremento en la productividad por reutilización

Introducción y Conceptos

Programación Orientada a Objetos

Conceptos fundamentales

- Clase
- Herencia
- Objeto
 - ☐ Método
 - ☐ Propiedad o atributo

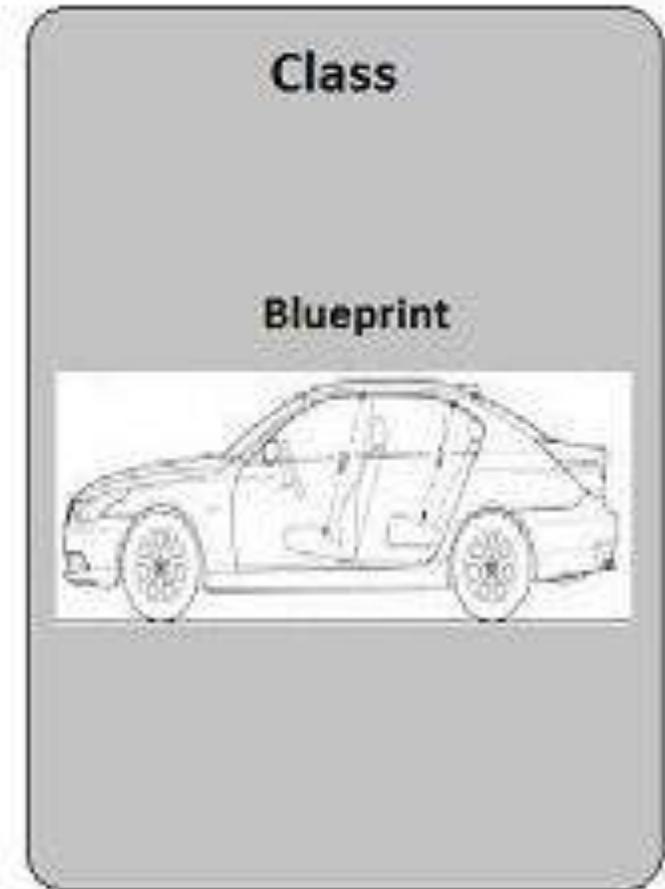


[Esta foto de Autor desconocido está bajo licencia CC BY-NC-ND](#)

Clase

- Es el esquema o plantilla, que representa las características (atributos) y acciones (Métodos) que realizarán los objetos creados a partir de esta.

<i>Vehiculo</i>
pasajeros : int
capacidadGL : float
kilometros : float
glDeposito : float
agregarCombustible(float galones)



[Esta foto](#) de Autor desconocido está bajo licencia [CC BY-SA](#)

Objeto

- Elemento creado a partir de una clase, a lo que se le llama instancia de una clase.
- Este cuenta con las características (Propiedades) y funcionalidades (métodos) definidos por la clase.



Herencia

Es la capacidad que tienen la clase mediante la cual se relaciona con otras clases, heredando atributos y operaciones



Esta foto de Autor desconocido está bajo licencia [CC BY-NC](https://creativecommons.org/licenses/by-nc/4.0/)

Constructores



Esta foto de Autor desconocido está bajo licencia [CC BY-NC-ND](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Constructores

Es una subrutina (método), que tiene por objetivo inicializar una nueva instancia de la clase, asignando valores iniciales al objeto instanciado.

En Java este método es invocado automáticamente al crear un objeto. Y una de sus características es que este lleva el mismo nombre de la clase, y también Java asume un constructor vacío por defecto.

Constructores – Tipos

De manera general existen 2 tipos de constructores

Constructor por defecto:

- Este no tiene parámetros en su definición y en caso de no definir uno específico dentro de la clase Java supondrá uno vacío.

Constructor parametrizado:

- Es un constructor que define parámetros para la inicialización del objeto, y se pueden definir cuantos sean necesarios siempre que no tengan el mismo número de parámetros o sus tipos de datos.

Constructores – Tipos

Ejemplo:

```
class Estudiante{
    private String nombre;

    //Constructor por defecto
    public Estudiante(){
        //Contenido de constructor
    }

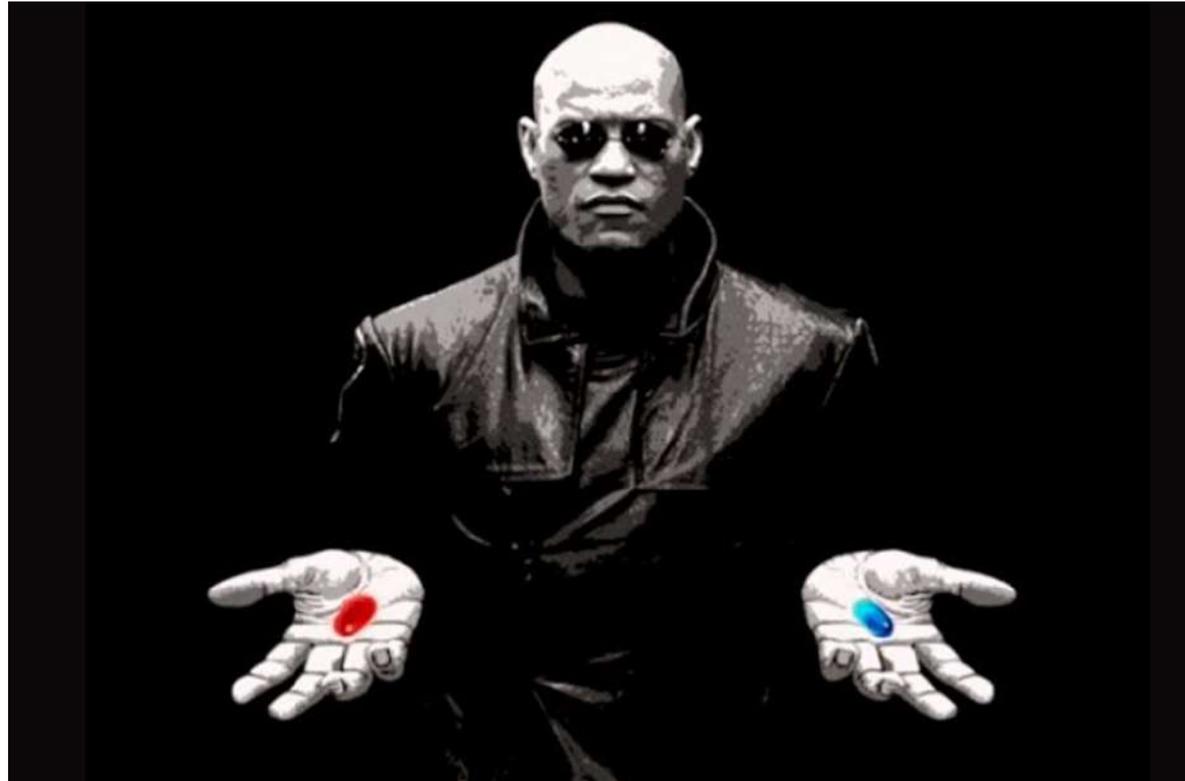
    //Constructor Parametrizado
    public Estudiante(String parametroNombre)
    {
        this.nombre=parametroNombre;
    }
}
```

Ejercicio

Sobre el ejercicio del estudiante, implemente:

- Un constructor por defecto que le asigne como nombre “Nuevo estudiante”, y que lo inscriba en su carrera por defecto.
- Cree mediante constructores parametrizados, defina los atributos de un nuevo estudiante.

ENCAPSULAMIENTO



[Esta foto](#) de Autor desconocido está bajo licencia [CC BY-NC-ND](#)

Encapsulamiento

Este concepto se refiere a la agrupación mediante métodos que operan con los datos, evitando la interacción directa con algunos de los componentes de un objeto

En java este concepto se asocia con los modificadores de acceso, mediante los cuales se especifica la accesibilidad de un miembro o tipo de la clase. (Estos están sujetos a los lenguajes de programación)

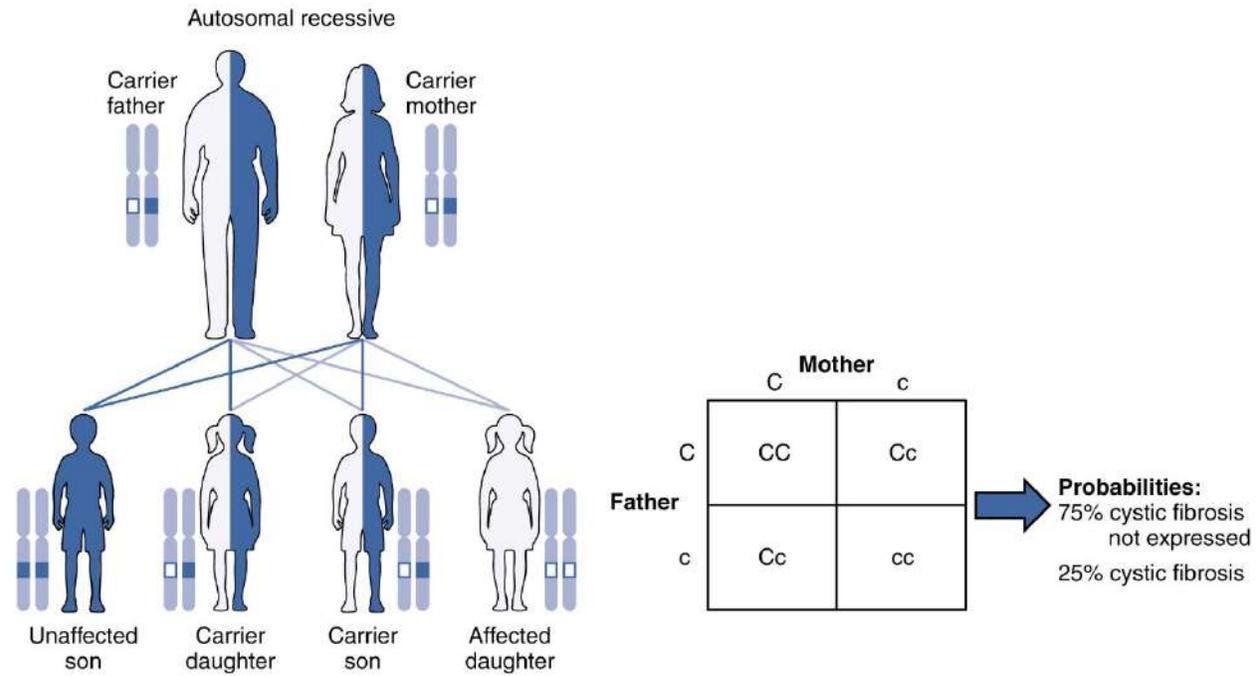
Modificadores de acceso

El propósito de utilizar los modificadores de acceso es evitar el mal uso de los miembros (atributos y métodos) de una clase, y principalmente se tienen modificadores de acceso:

- Público (public)
- Privado (private)
- Protegido (protected)
- Por defecto (default) - No se requiere palabra clave

<https://javadesdecero.es/poo/modificadores-de-acceso/>

Herencia



Herencia

Es uno de los conceptos mas importantes de la programación orientada a objetos.

Se basa en la premisa de reutilización de código que ya se ha implementado en una clase.

También se le entiende como **Jerarquías de clases** o generalización (herencia), Y esta basado en el principio por el cual el conocimiento de una categoría más general es aplicable a una categoría más específica se denomina generalización, y su implementación en POO se denomina herencia.

Herencia – Implementación

```
//clase padre  
  
class Persona {  
  
    // atributos de la clase padre  
    String nombre;  
    int edad;  
    int telefono;  
  
}
```

```
//Clase hija  
class Estudiante extends Persona{  
    //atributo de la clase hija  
    String Carrera;  
  
}
```

Lectura de estudio



Realice la lectura de los 2 primeros capítulos del libro en el siguiente vinculo, como complemento a los contenidos de la clase:

http://www.cua.uam.mx/pdfs/revistas_electronicas/libros-electronicos/2016/2intro-poo/programacion_web.pdf