

Punto 1 de parcial de poo Juan Correa "20251005092"

1.1 ¿En la programación orientada a objetos qué es el constructor y qué función cumple?

Un constructor es un método especial dentro de una clase que se ejecuta automáticamente cuando se crea un nuevo objeto de esa clase.

Su función principal es inicializar los atributos del objeto (darles valores iniciales) y preparar cualquier recurso necesario para su funcionamiento.

1.2 ¿En la programación orientada a objetos qué es un objeto y qué función cumple?

Un objeto es una instancia de una clase, es decir, una representación concreta de esa clase en memoria.

Cumple la función de encapsular datos (atributos) y comportamientos (métodos) que pertenecen a una entidad del mundo real o lógico.

1.3 ¿En la programación orientada a objetos qué es una clase y qué función cumple?

Una clase es una plantilla o modelo que define cómo serán los objetos: qué datos tendrán (atributos) y qué acciones podrán realizar (métodos).

Cumple la función de organizar y estructurar el código, permitiendo crear múltiples objetos similares sin repetir código.

1.4 ¿Qué diferencias destacadas se aprecian entre la OOP y la programación estructurada (secuencial)?

Característica	Programación estructurada	Programación orientada a objetos (OOP)
Unidad básica	Funciones o procedimientos	Clases y objetos
Enfoque	En las acciones (qué hace el programa)	En los datos y sus interacciones
Reutilización de código	Limitada, se basa en funciones	Alta, mediante clases y herencia
Encapsulamiento	No existe formalmente	Sí, protege los datos del acceso directo

Característica	Programación estructurada	Programación orientada a objetos (OOP)
Mantenimiento	Más complejo en proyectos grandes	Más fácil, por modularidad
Ejemplo	Programas en C	Programas en Java, Python, C++

1.5 ¿Para qué y por qué se usan modificadores de acceso?

Los modificadores de acceso se usan para controlar el nivel de visibilidad de los atributos, métodos y clases dentro de un programa.

Su propósito es proteger la información (encapsulamiento) y evitar un uso indebido de los datos internos de los objetos.