INTRODUCCION AL PARADIGMA DE OBJETOS.

**MODULO I**

**Paradigma**:

Teoria o conjuntos de teorias que sirven de modelo a seguir para resolver problemas.

La POO define los programas por medio de “**clases de objetos**”

**Los objetos combinan:**

- Identidad (nombre de la clase)

- Estado (caracteristicas)

- Comportamiento ( acciones )

Los objetos conviven en un programa y se comunican a traves de mensajes.

Ej de lenguajes de POO

- Java

- PHP

- Phyton

- C#

**Ventajas de la PPO**

- Fomenta la reutilizacion del codigo

- Crear sistemas complejos

- Semejanza al mundo real

- Agiliza el desarrollo de sofware

- Facilita el trabajo en equipo.

**Conceptos basicos de la PPO**

**Objeto**: Entidad provista de un conjunto de atributos (datos) y comportamiento (metodos)

**Clase:** Es el modelo en que se definen la propiedades y comportamiento de un tipo de objeto. (molde) Se usa para ir creando objetos. Esta compuesta por atributos (datos), tambien llamadas variables de instancia o variables miembro que nos indican en que estado se encuentra el objeto. Tambien se compone de metodos (funcones o procedimientos) que indican cual es el comportamiento.

Los atributos de la clase son privados al objeto y su acceso se realiza mediante los metodos definidos para manejar esos atributos.

**Metodo (operacion):** Son acciones contenidas por el objeto, que se asignan en la clase. Determina el comportamiento que puede tener un objeto, el cual fue referido en la clase. Es un algoritmo asociado a un objeto o una clase de objetos que se ejecuta tras la recepcion de un mensaje. Son generalmente verbos. Pueden admitir variables como parametros de entrada y retornar valores de salida o retorno. Los metodos u operaciones modifican el estado de los objetos.

Valores de **entrada** (**datos**): Son valores enviados a un metodo para realizar la operación.

Valores de **salida** o retorno (**informacion**): Es el valor devuelto por un operación luego de realizar un procesamiento. Todos los valores de salida deben tener un tipo de dato asociado. Solo devuelven 1 solo objeto.

**Mensaje:** Es la comunicación hacia un objeto para que este ejecute uno de sus metodos para obtener un resultado.

**Propiedad o atributo:** Es un tipo de dato que define a nuestro objeto. Son las caracteristicas y normalmente cuentan con un dato asociado. Siempre seran sustantivos que haran referencia a nombres, tamaños formas y estados. Las clases definen los atributos y los objetos los completan !

**Estado interno:** Segun sus valores en un momento dado representan el estado de ese objeto en ese momento. Ese estado puede ser modificado a traves de un metodo definido en la misma clase a cual pertenece el objeto.

**Componentes de un objeto:**

- Identidad

- Atributos

- Metodos

- Relaciones

**Representacion de un objeto:**

Va a ser representado por medio de una entidad compuesta por sus atributos y funciones.

**Comunicacion entre objetos.**

Emisor le pide algo al receptor y puede que obtenga una respuesta o no.

**Caracteristicas de la PPO**

- **Abstraccion** (nos permite seleccionar ciertas caracteristicas del objeto atributos o comportamientos según necesitemos)

- **Encapsulamiento** Nos permite reunir todos los elementos que pertenecen a una entidad particular.

- **Herencia** Las clases se relacionan entre si y forman una jerarquia de clasificacion. Los objetos heredan la propiedades y comportamiento de todas las clases a las que pertenecen.

- **Polimorfismo** Se pueden definir clases diferentes en la misma jerarquia de herencias que tienen metodos denominados de forma identica pero se comportan de manera distinta.

**MODELOS**

Los Modelados previos de los sistemas de informacion sirven de intermedio para poder entender el problema y la solucion en la orientacion a objetos

**Diagrama de clases (el mas importante)**

1- Identificar clases y objetos del dominio del problema.

2- Definir el comportamiento de los objetos.

3- Asociar estaticamente los objetos.

4- Definir la jerarquia de herencia de clases

**DIAGRAMA UML** es una herramienta para diagramas de modelado que incluye casos de usos, actividades, iteracion, estados e implementacion. (*drawio*: sofware libre para diagramar)

**MODULO II**

**RELACOINES ESTRUCTURALES**

Son relaciones que se dan entre los objetos de diferentes clases.

**Asociacion** Es una relacion estructural que especifica que los objetos de un elemento estan conectados con los objetos de otro elemento. Dada una asociacion entre dos clases se pude navegar de un objeto de una a un objeto de otra.

**Multiplicidad** de una asociacion determina cuantos objetos de cada tipo intervienen en la relacion. Es el numero de instancias de una clase que se relaciona con una instancia de otra clase. Cada asociacion tiene **dos multiplicadores** (uno en cada extremo).

Ej 1 ------> 1,N

**Relacion Simple** 1 – 1 (uno a uno)

**Relaciones multiples** 1 – N (uno a muchos)

**SOBRECARGA DE OPERACIONES**

Es la aparicion de metodos dentro de una misma clase que se llaman igual, pero realizan acciones diferentes.

Nos permite utilizar el mismo nombre para dos operaciones distintas, que aceptan distinta cantidad y/o tipos de parametros y tienen distinta funcionalidad. El tipo de devolucion no puede ser el mismo. El compilador distingue estas operaciones según el tipo y cantidad de parametros.

Ej: class automovis {

void acelerar () { velocidad += 10; }

void acelerar (int km) { velocidad += km; }

void acelerar (int km, boolean nitro); {

if (nitro) { acelerar(km\*2); }

else { acelerar(km); }

}

**VISIBILIDAD**

Determina el alcance que tiene una clase. Es desde donde puedo acceder a una clase, atributo u operación.

El alcance es fundamental en la POO, ya que permite definir el encapsulamiento de la informacion y el comportamiento. Los metodos, atributos y clases deben tener un solo alcance, tambien llamado **tipo de acceso.**

**Modificadores de visibilidad**

**- defaul:** Acceso por defecto. Un elemento s

**- public**

**- private**

**- protected**