**CONCEPTOS BÁSICOS DE POO**

* Diseñar un mapa conceptual o mapa de ideas, sobre los conceptos que se le enumerarán a continuación. Las definiciones de dichos conceptos deben ser redactados con sus propias palabras.

DEFINICIONES

Objetos

Los objetos son como "instancias" de una clase en la programación. Son como representaciones de entidades reales o ideas abstractas en el mundo. Cada objeto tiene su propio estado y comportamiento, lo cual le da una identidad única y la capacidad de interactuar con su entorno.

Propiedades o atributos

Las propiedades o atributos son como las características o atributos que posee un objeto. Estas propiedades definen las características específicas de cada instancia y pueden ser variables o constantes. Funcionan como variables que almacenan el estado del objeto, permitiendo representar sus propiedades distintivas.

Estado

El estado de un objeto se refiere al conjunto de valores almacenados en sus atributos en un momento determinado. Es una representación de la condición o situación actual del objeto. El estado de un objeto puede cambiar a lo largo de la ejecución del programa, el estado es dinámico y refleja la información actualizada de un objeto en un determinado momento.

Métodos

Son como funciones asociadas a una clase u objeto en la programación. Representan el comportamiento o las acciones que un objeto puede realizar. Estas funciones encapsulan la lógica necesaria para realizar operaciones específicas en relación con un objeto, ya sea manipulando sus atributos, realizando cálculos o interactuando con otros objetos

Clase

Una clase en programación es una plantilla para objetos. Define estructura y comportamiento. Atributos y métodos se heredan. Atributos son características, métodos son acciones. Objetos comparten propiedades y ejecutan acciones. Clase: plantilla para objetos, establece estructura, comportamiento y herencia de atributos y métodos.

Instanciación

La instanciación es el proceso mediante el cual se crea una instancia u objeto de una clase en programación. Cada instancia representa una entidad única y puede ser manipulada independientemente de otras instancias de la misma clase.

Evento

Un evento en programación es una acción o suceso significativo que ocurre dentro de un programa. Puede ser una interacción del usuario, un cambio de estado, una entrada de datos, entre otros. Los objetos pueden responder a eventos ejecutando los métodos asociados a ellos.

Herencia

La herencia es un mecanismo que permite crear nuevas clases basadas en clases existentes. Este enfoque facilita la reutilización de código y la especialización de comportamiento.

Abstracción

La abstracción es el proceso de identificar las características esenciales de un objeto y omitir los detalles irrelevantes. Permite crear modelos y simplificar la representación de objetos complejos.

Encapsulamiento

El encapsulamiento es el principio de ocultar los detalles internos de un objeto y exponer solo la interfaz necesaria para interactuar con él. Proporciona protección a los atributos y controla el acceso a ellos, evitando modificaciones no autorizadas.

Ocultamiento

El ocultamiento es la práctica de ocultar los detalles de implementación de un objeto. Los detalles internos no son accesibles desde fuera del objeto, lo que brinda mayor seguridad.

Modificadores de acceso

Los modificadores de acceso son palabras clave que definen el alcance y la visibilidad de los atributos y métodos en la programación orientada a objetos, brindan control sobre la accesibilidad y el nivel de encapsulamiento de los miembros de una clase. (Public, Protected, Private),

Modularidad

Modularidad es el principio de dividir un programa en módulos o unidades funcionales independientes, cada módulo tiene una responsabilidad específica y puede ser desarrollado y mantenido de forma aislada, por ejemplo, el encapsulamiento y el ocultamiento son considerados modularidad.

Polimorfismo

El polimorfismo es la capacidad de objetos de diferentes clases de responder al mismo mensaje o ejecutar el mismo método. Por ejemplo, enviar la misma notificación a diferentes destinos.

A picture containing text, screenshot, rectangle, design

Description automatically generated

* Mediante dibujos o gráficas, describa un ejemplo de cada una de las siguientes características (no pueden ser los mismos ejemplos expuestos en clase):

a. Clase y objetos: instanciación, atributos, métodos

b. Clases: herencia

A picture containing text, screenshot, font, number

Description automatically generated

Descripción de los ejemplos:

Ejemplo 1:

Clase: Mascota

Atributos: Nombre, Peso, Raza

Métodos: Obtener\_Nombre, Obtener\_Peso, Obtener\_Raza

Instancias: Firulais, Pequitas, Peluquín

En este ejemplo, se define una clase llamada "Mascota" que tiene tres atributos: Nombre, Peso y Raza. Los métodos "Obtener\_Nombre", "Obtener\_Peso" y "Obtener\_Raza" son utilizados para obtener los valores de los respectivos atributos. Luego se crean tres instancias de la clase "Mascota" llamadas Firulais, Pequitas y Peluquín. Cada instancia representa una mascota individual con un nombre, peso y raza específicos.

Ejemplo 2:

Clase: Mascota

Atributos: Nombre, Peso, Raza

Métodos: Obtener\_Nombre, Obtener\_Peso, Obtener\_Raza

Herencia: Perro, Gato, Conejo

En este ejemplo, también se tiene una clase llamada "Mascota" con los mismos atributos y métodos que el ejemplo anterior. Sin embargo, se introduce la herencia con las clases derivadas: Perro, Gato y Conejo. Estas clases heredan los atributos y métodos de la clase base "Mascota". Esto significa que los objetos de las clases derivadas también tendrán nombre, peso y raza, y podrán utilizar los métodos para obtener esos valores específicos de cada instancia. La herencia permite establecer una relación jerárquica entre las clases y compartir comportamientos comunes mientras se agrega funcionalidad adicional específica en las clases derivadas.