

Taller de conceptos y principios de programacion orientada a objetos

OOP/POO

GA4-220501095-AA2-EV01



Isidro J Gallardo Navarro

Ficha:3070299

2025

**Tecnología en Análisis y Desarrollo de
Software.**

ADSO

Puntos a desarrollar:

1. Presenta el glosario de terminología utilizada en la POO.
2. Explica las características, los principios o pilares Básicos de la POO.
3. Definir y aplica correctamente todos los conceptos claves de la POO.
4. Aplicar las normas básicas de presentación de un documento escrito.

Introducción

Este paradigma de la programación se enfoca en la representación del mundo real en el internet de las cosas en el presente escrito tendrá como propósito exponer los principales términos y los pilares de este paradigma, como su nombre le indica ésta manera de programar se enfoca en los elementos que hacen parte de un programa, y de esta manera los divide a manera de subprogramas o en algunos casos se refieren como modular, dentro del estilo de programación orientada a objetos se identifican tres elementos fundamentales los cuales son;

Objetos,
Clases,
Método.

Dentro de los sistemas informáticos los objetos que creamos lo hacemos a través de ideas que requiere de una estructura, una de las formas para acceder al mundo de las ideas según Platón es el Lenguaje el cual podemos emplear para modelar las ideas, como ejemplo de esto tenemos una tapa de una botella, la cual tiene forma circular, contiene la idea que tenemos sobre lo que es un círculo, pero no es exactamente un círculo, la Tapa de la Botella me hace conocer la idea del círculo a través del conocimiento sensible luego al inteligible y a su vez lo inteligible nos hace criticar lo sensible.

Para entender la idea de clase dentro de los sistemas informáticos orientados objetos hago un símil a un molde del cual se obtendrán varios objetos los cuales compartirán características básicas dentro de esta se definen los atributos y métodos que comprenden los objetos;

"Mirad vuestro perro, ¡que tranquilo y contento está! Millares de perros han muerto antes que este viniese a la vida, pero la desaparición de todos aquellos no ha tocado la idea del perro.

Esta idea no se ha oscurecido por su muerte. he aquí porque vuestro perro está tan fresco, tan animado por fuerzas juveniles cómo si este fuera su primer día y no hubiese de tener termino. A través de sus ojos brilla el principio indestructible que hay en el Archoeus. ¿Qué es pues lo que la muerte ha destruido a través de millaeres de años?, no es el perro; ahí está, delante de vosotros, sin haber sufrido detrimento alguno. Sólo su sombra, su figura, es lo que la debilidad de nuestro conocimiento no puede percibir sino en el tiempo."
Schopenhauer, A. (28)

En el anterior ejemplo podemos ver como el autor propone la idea de perro independiente del perro en si, también como se le atribuye a dicho perro un ciclo de vida, principios que se

comparten dentro de la programación orientada a objetos.

El Concepto de "Idea del Perro" = Clase

El texto habla de la "idea del perro" que permanece intacta a pesar de que "millares de perros han muerto". Esta "idea" es análoga al concepto de clase en POO:

La clase Perro es la definición abstracta, el molde o plantilla

Los perros individuales son las instancias u objetos específicos

Cuando un perro muere (un objeto es destruido), la clase Perro permanece intacta

Instanciación y Mortalidad de Objetos

"Millares de perros han muerto antes que este viniese a la vida" refleja perfectamente el ciclo de vida de los objeto

El "Principio Indestructible" = Herencia y Polimorfismo

El texto menciona que "a través de sus ojos brilla el principio indestructible". En POO, esto se relaciona con:

Herencia: Los rasgos esenciales se transmiten de la clase padre a las instancias

Polimorfismo: Diferentes perros pueden comportarse de manera similar manteniendo su esencia

Abstracción vs Implementación

"Sólo su sombra, su figura, es lo que la debilidad de nuestro conocimiento no puede percibir sino en el tiempo" es una reflexión profunda sobre:

Abstracción: La idea pura del perro (la clase) existe fuera del tiempo

Implementación: Las instancias concretas (objetos) existen en el tiempo y el espacio.

Con el anterior ejemplo espero haber dado una idea general sobre el POO, ahora sera necesario enumerar y definir dichos conceptos Principios de orientación a objetos:

Encapsulación

Clase

Abstracción

Herencia

Polimorfismo

Composición y

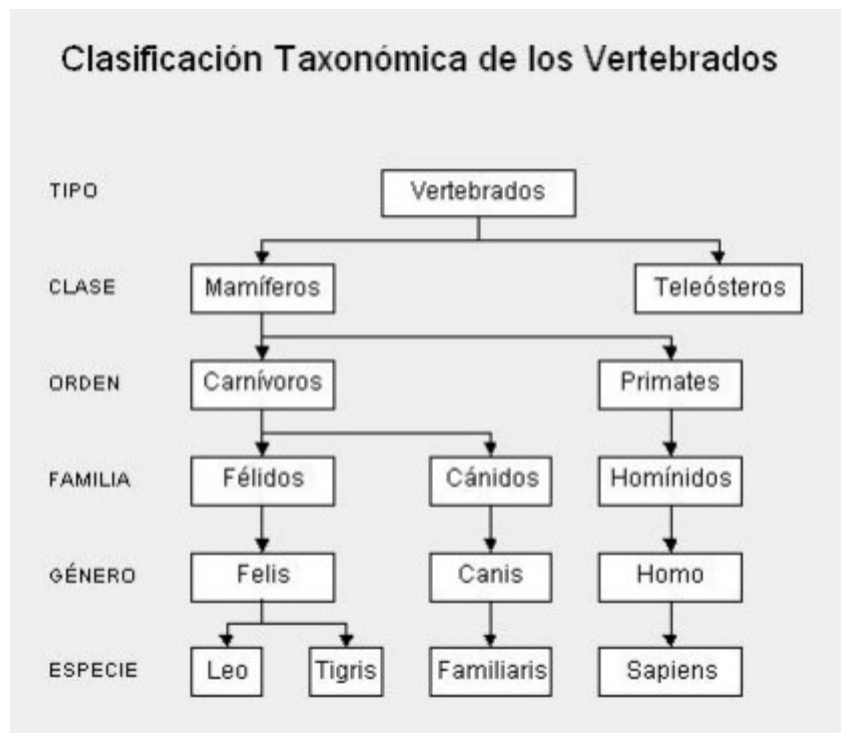
agregación

Encapsulamiento: el objeto esconde sus datos de los demás objetos y permite el acceso a los datos mediante sus propios métodos. Esto recibe el nombre de ocultamiento de información. El encapsulamiento evita la corrupción de los datos de un objeto. Si todos los programas pudieran tener acceso a los datos de cualquier forma que quisieran los usuarios, los datos se podrían corromper o utilizar de manera incorrecta. El encapsulamiento protege los datos del uso arbitrario y no pretendido.

El encapsulamiento oculta los detalles de su implantación interna a los usuarios de un objeto. Los usuarios se dan cuenta de las operaciones que pueden solicitar del objeto, pero desconocen los detalles de cómo se lleva a cabo la operación. Todos los detalles específicos de los datos del objeto y la codificación de sus operaciones están fuera del alcance del usuario.

Clase: una clase es una plantilla que define las variables y métodos comunes entre unos objetos específicos, de ella pueden derivar varios objetos del mismo tipo, cuando se define una clase, se declara su forma y naturaleza exacta. Esto se hace especificando los datos que contiene la clase y el código que opera sobre ellos.

Aunque las clases sencillas pueden contener sólo código o datos, la mayoría de las clases reales contienen las dos.



Métodos

La mayor parte de las veces es necesario utilizar métodos para acceder a las variables de instancia definidas en una clase. De hecho los métodos definen la interfaz de la mayoría de las clases. Esto permite que la clase oculte sus estructuras de datos internos detrás de un conjunto de métodos.

Encapsulamiento

El empaquetamiento de los datos y métodos se conoce como encapsulamiento. El objeto esconde sus datos de los demás objetos y permite el acceso a los datos mediante sus propios métodos. Esto recibe el nombre de ocultamiento de información. El encapsulamiento evita la corrupción de los datos de un objeto.

Si todos los programas pudieran tener acceso a los datos de cualquier forma que quisieran los usuarios, los datos se podrían corromper o utilizar de manera incorrecta. El encapsulamiento protege los datos del uso arbitrario y no pretendido.

El encapsulamiento oculta los detalles de su implantación interna a los usuarios de un objeto. Los usuarios se dan cuenta de las operaciones que pueden solicitar del objeto, pero desconocen los detalles de cómo se lleva a cabo la operación. Todos los detalles específicos de los datos del objeto y la codificación de sus operaciones están fuera del alcance del usuario.

Herencia

La Herencia es el mecanismo por el cual se crean nuevos objetos definidos en términos de objetos ya existentes. Una subclase hereda propiedades de su clase padre, también llamada superclase. Una subclase también puede heredar propiedades de varias superclases. La subclase también puede tener sus propios métodos e incluso sus mismos tipos de datos.

Polimorfismo

También llamado sobrecarga de métodos. En Java es posible definir dos o más métodos dentro de la misma clase que tengan el mismo nombre, pero con sus listas de parámetros distintas. Cuando ocurre esto, se dice que los métodos están sobrecargados y a este proceso se le denomina sobrecarga de métodos.

References

Gallardo Navarro, IJ. (2021). evidencia001ObjClass.

Moreno, F. (1999). Introduccion a la OOP (pimera ed., Vol. 1).

Principio del Analisis y Dise~o orientado a Objetos Utilizando el Estandar UML (segunda ed.). (2006). Felix Ramon Triana Gaitan.

schopenhauer, a. (n.d.). El Amor Las Mujeres Y la Muerte.