

INFORMATICA PARA INGENIEROS PARADIGMA ORIENTADO A OBJETOS

LECTURAS CUARTA SESION

<https://damian02.wordpress.com/objetos-poo/>

https://www.youtube.com/watch?v=-6BYa_x_QA0

La programación orientada a objetos surge en 1969, aunque su mayor desarrollo ha sido en la década de los 80's y continúa su desarrollo en la siguiente década.

Objetos (P.O.O)

OBJETOS:

-Conceptos

-Objeto: entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos). Se corresponde con los objetos reales del mundo que nos rodea, o a objetos internos del sistema (del programa). Es una instancia a una clase.

-objeto: En el paradigma de programación orientada a objetos (POO, o OOP en inglés), un objeto es la unidad individual que en tiempo de ejecución realiza las tareas de un programa. Estos objetos interactúan unos con otros, en contraposición a la visión tradicional en la cual un programa es una colección de subrutinas (funciones o procedimientos), o simplemente una lista de instrucciones para el computador. Cada objeto es capaz de recibir mensajes, procesar datos y enviar mensajes a otros objetos de manera similar a un servicio.

En el mundo de la programación orientada a objetos (POO), un objeto es el resultado de la instanciación de una clase. Una clase es el anteproyecto que ofrece la funcionalidad en ella definida, pero ésta queda implementada sólo al crear una instancia de la clase, en la forma de un objeto. Por ejemplo: dado un plano para construir sillas (una clase de nombre clase_silla), entonces una silla concreta, en la que podemos sentarnos, construida a partir de este plano, sería un objeto de clase_silla. Es posible crear (construir) múltiples objetos (sillas) utilizando la definición de la clase (plano) anterior. Los conceptos de clase y objetos son análogos a los de tipo de datos y variable, es decir, definida una clase podemos crear objetos de esa clase, igual que disponiendo de un determinado tipo de dato (por ejemplo el tipo entero), podemos definir variables de dicho tipo.[1]

-ejemplos:

Entonces, en vez de programar una clase, para estar orientados a objetos en JS definimos un prototipo. Las variables dentro de este serán las propiedades, y las funciones serán los métodos:

```
[Objeto = Prototipo]{  
  [ Propiedad = Variable ]  
  
  [ Método = Funcion ]  
  
}  
[6]
```

Objetos

Un objeto es una cosa tangible, algo a que se puede aprehender intelectualmente o algo hacia lo que se puede dirigir una acción o pensamiento.

Un objeto representa un item individual e identificable, o una entidad real o abstracta, con un papel definido en el dominio del problema

Un objeto tiene:

Estado

Comportamiento

Identidad

La estructura y el comportamiento de objetos similares se definen en sus clases comunes. El término objeto y ejemplo (instance) de una clase son intercambiables.

Estado de un objeto

El estado de un objeto abarca todas las propiedades del objeto, y los valores actuales de cada una de esas propiedades. Las propiedades de los objetos suelen ser estáticas, mientras los valores que toman estas propiedades cambian con el tiempo.

El hecho de que los objetos tengan estado implica que ocupan un espacio, ya en el mundo físico , ya en la memoria del ordenador.

El estado de un objeto está influido por la historia del objeto.

No deben confundirse los objetos, que existen en el tiempo, son mutables, tienen estado, pueden ser creados, destruidos y compartidos..., con los valores (los asignados a una variable, por ejemplo) que son cantidades con las propiedades de ser atemporales, inmutables.

El estado de un objeto representa el efecto acumulado de su comportamiento.

Identidad de un objeto

Identidad es la propiedad de un objeto que lo lleva a distinguirse de otros.

COMPORTAMIENTO DE UN OBJETO

Comportamiento es como un objeto actúa y reacciona, en términos de sus cambios de estado y de los mensajes que intercambia.

El comportamiento de un objeto representa su actividad externamente visible y testable. Son las operaciones que una clase realiza (llamadas también mensajes) las que dan cuenta de como se comporta la clase. Por operación se denota el servicio que una clase ofrece a sus clientes. Un objeto puede realizar cinco tipos de operaciones sobre otro, con el propósito de provocar una reacción:

Modificador: altera el estado de un objeto.

Selector: accede al estado de un objeto, sin alterarlo.

Iterador: permite a todas las partes de un objeto ser accedidas en un orden.

Constructor: crea un objeto y/o inicializa su estado.

Destructor: libera el estado de un objeto y/o destruye el objeto.

C++ soporta, además de las operaciones, subprogramas libres. En la terminología de C++ las operaciones que un cliente puede realizar sobre un objeto se declaran como funciones miembro.

Relaciones entre objetos

Las relaciones entre objetos abarcan las operaciones, resultados y suposiciones que unos hacen sobre los otros.

Links Son conexiones físicas o conceptuales entre objetos. Denota la asociación específica por la que un objeto (cliente) usa o solicita el servicio de otro objeto (servidor). El paso de mensajes entre objetos los sincroniza.

Agregaciones Denota relaciones todo/parte, con capacidad para gobernar desde el todo las partes. Es equivalente a la relación "tener un". El todo puede contener a la parte.

Agregación es conveniente en las ocasiones en que el encapsulamiento de las partes es prioritario. Si se requiere que las relaciones entre objetos esten vagamente acopladas, se utilizan links.

Clases

Una clase es un conjunto de objetos que comparten una estructura y comportamiento comunes.

Clase representa una abstracción, la esencia que comparten los objetos.

Un objeto es un ejemplo de una clase.

Un objeto no es una clase, y una clase no es un objeto (aunque puede serlo, p.e. en Smalltalk).

Las clases actuan como intermediarias entre una abstracción y los clientes que pretenden utilizar la abstracción. De esta forma, la clase muestra:

visión externa de comportamiento (interface), que enfatiza la abstracción escondiendo su estructura y secretos de comportamiento.

visión interna (implementación), que abarca el código que se ofrece en la interface de la clase.

LECTURA 2

La programación orientada a objetos es un enfoque de programación que combina datos y acciones asociadas (métodos) en estructuras lógicas (objetos). Este enfoque aumenta la capacidad para administrar la complejidad del software, lo cual resulta especialmente importante cuando se desarrollan y mantienen aplicaciones y estructuras de datos de gran tamaño. (MATLAB, 2018).

La programación Orientada a objetos (POO, u OOP según sus siglas en inglés) es un paradigma de programación que usa objetos en sus interacciones, para diseñar aplicaciones y programas informáticos.

Está basada en varias técnicas, incluyendo herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento. Su uso se popularizó a principios de la década de 1990. En la actualidad, existe una gran variedad de lenguajes de programación que soportan la orientación a objetos. (Ángel Roldán, 2017).

La POO es una forma especial de programar, este paradigma de programación es cercano a como expresamos las cosas en la vida real en nuestro día a día. Con la POO tenemos que aprender a pensar las cosas de una manera distinta para poder escribir nuestros programas en términos de objetos, propiedades y métodos. La POO tiene como principio que todo en la vida es un objeto

programable, entonces para empezar a programar con este paradigma tendríamos que empezar con desarrollar nuestro pensamiento basado en objetos.

¿Cómo pensar en objetos?

Pensar en términos de objetos es pensar de manera natural como lo haríamos en la vida real. Por ejemplo al pensar en un coche para modelarlo en un esquema de POO. Diríamos que el coche es el elemento principal que tiene una serie de características, como podrían ser el color, el modelo o la marca. Además tiene una serie de funcionalidades o métodos asociados al mismo, como pueden ser ponerse en marcha, parar o acelerar.

En un esquema POO el coche sería el objeto, las propiedades serían las características como el color o el modelo y los métodos serían las funcionalidades asociadas como ponerse en marcha o parar.

Los programas Orientados a objetos utilizan muchos objetos para realizar las acciones que se desean realizar y ellos mismos también son objetos, por ejemplo un taller de coches, el taller de coches será un objeto que utilizará objetos coche, herramienta, mecánico, recambios, etc.

Algunas de las ventajas de la programación orientada a objetos son las siguientes:

- Los componentes se pueden reutilizar.
- Facilidad de mantenimiento y modificación de los objetos existentes.
- Una estructura modular clara se puede obtener, la cual no revelará el mecanismo detrás del diseño.
- Se proporciona un buen marco que facilita la creación de rica interfaz gráfica de usuario aplicaciones (GUI).
- Se acopla bien a la utilización de bases de datos, debido a la correspondencia entre las estructuras.

Desventajas

- Limitaciones del programador: Es posible el programador desconozca algunas características del paradigma y de hecho siga utilizando el paradigma estructurado.

- No hay una forma única de resolver los problemas. Esto puede llevar a que diferentes interpretaciones de la solución planteada emerjan.
- Se requiere una documentación amplia para determinar la solución planteada.

La programación orientada a objetos es una forma especial de pensar, nos permite modelar nuestra realidad dentro de un entorno de programación, partiendo de la premisa de que todo en esta vida es un objeto programable.