

Explorando las bases de la programación orientada a objetos

-Object-oriented programming

Angel gael rodriguez gonzalez
Universidad politécnica de victoria, Mexico.
2330145@upv.edu.mx



nombre de la maestra: Sheyla Maleny Silva Gonzalez

nombre de la escuela: Universidad Politécnica de Victoria

resumen:

En este artículo habla sobre los pilares de la programación orientada a objetos, ejemplos y conceptos, así como también mencionó los antecedentes de la programación orientada a objetos

abstract:

In this article I mention about the pillars of object-oriented programming, examples, and concepts, as well as mentioning the background of object-oriented programming

1. introducción:

La programación orientada a objetos (POO) es un enfoque revolucionario para el diseño y la construcción del software. A diferencia de la programación tradicional, que se centra en la lógica y las funciones, la POO se centra en los objetos y las interacciones entre ellos.

En el artículo hablaré sobre conceptos sobre los fundamentos de la POO, como clases, herencias, el encapsulamiento y el poliformismo, así como estos conceptos pueden mejorar la calidad y la eficiencia del software.

Este artículo se busca ampliar tus conocimientos sobre los fundamentos de la programación y te tratará de dar una comprensión sobre la POO y el desarrollo de software moderno

con lo anterior, los sistemas orientadas a objetos si tienen un papel determinantes, lo importante no es investigar si un sistema está orientado a objetos o no, sino en qué forma está orientada a objetos y en que manera esta proporcionada

conceptos básicos de la programación orientada a objetos

Marco Celedon(1997) dice que algunas de las ideas fundamentales que subyacen en la tecnología orientada a objetos son las siguientes:

- Objetos
- Métodos
- Herencia
- Encapsulado
- Polimorfismo
- Clases

Definición de objeto.

La representación en software OO del objeto es entonces una colección de tipos de datos y métodos. En software OO un objeto es cualquier cosa, real o abstracta, acerca de la cual almacenamos datos y los métodos que controlan dichos datos.

A continuación mencionaremos algunos conceptos de Objeto en la metodología de programación orientada a objetos:

- Un objeto es una entidad capaz de almacenar un estado (información) y que ofrece varias operaciones (comportamiento) que nos permite examinar o afectar su estado.
- Un objeto es el elemento básico de la programación orientada a objetos. Un objeto es aquel que es capaz de retener información y sabe cómo llevar a cabo sus operaciones.
- Un objeto es un conjunto de variables de software y métodos relacionados

Rosiles, M. A. C. (2023). Programación orientada a objetos.

Antecedentes históricos

Marco Celedon(1997) menciona que la aparición de la programación orientada a objetos.

La aparición de la programación orientada a objetos resume la historia de la computación como un todo. Los primeros trabajos de la computación se remontan al final de la década de los 40 's, tenía que ver con lo que ahora conocemos como programación. Solo un par de años más tarde surgió un interés consciente sobre el diseño y del mismo modo fue la programación orientada a objetos fue la que llamó la atención primero .

Ten Dyke y Kunz (1989) afirman que en el año 1957 ya se habían utilizado técnicas fundamentales de orientación a objetos. El lenguaje

comienza con el lenguaje de simulación de sucesos discretos,. Los lenguajes de influencia intermedia incluyen el alphard y el CLU, desde entonces ha habido muchos lenguajes que se han inspirado en esos desarrollos y reclaman el título de “orientación a objetos”.

A medida que la ingeniería de software comenzó a madurar, el interés se desplazó hacia los métodos de diseño a objetos y las especificaciones o el análisis orientado a objetos. Las ventajas de la reutilizabilidad y de la extensibilidad se pueden aplicar en tanto los diseños y a las especificaciones como el código.

por lo anterior,el futuro a la programación orientada a objetos parece muy prometedor, dada la importancia que se tiene el encapsulamiento y el paso de mensajes, al reconocimiento de la reutilizabilidad como el problema clave de la programación, el análisis y el diseño

Rosiles, M. A. C. (2023). Programación orientada a objetos.

¿Qué es la programación orientada a objetos?

El modelo de programación orientado a objetos denominado POO expresa un programa como un conjunto de objetos que colaboran entre sí para realizar tareas, es un paradigma de programación que modula nuestro código para hacerlo más entendible y se basa en crear nuevos tipos de datos llamados objetos.

Trataremos de nuestros en dos partes, en atributos y métodos

Montero, E. L. P., & Pérez, F. D. M. H. (2019). La programación orientada a objetos facilidad para crear. *I+ T+ C-Research, Technology and Science*, 1(13), 96-100.

clases

Mediante la POO, a la hora de tratar un problema, podemos descomponerlo en subgrupos de partes relacionadas. Estos subgrupos pueden traducirse en unidades autocontenidas llamadas objetos. Antes de la creación de un objeto, se debe definir en primer lugar su formato general, su plantilla, que recibe el nombre de clase. Por ejemplo, pensemos que estamos intentando construir una aplicación de dibujo de cuadrados. Nuestro programa va a ser capaz de pintar cuadrados con

diferentes tamaños y colores: uno será rojo y de 3 centímetros de lado, otro verde y de 5 centímetros de lado, etc. Como podemos deducir, cada cuadrado está definido por dos características, lado y color, y contiene la operación dibujar. Desde el punto de vista de la POO, cada cuadrado se va a representar como un objeto de la clase 'Cuadrado' que es la que contiene las características y operaciones comunes de los objetos.

Rodríguez, R., Sosa, E., & Prieto, Á. (2004). Programación orientada a objetos

Atributos

Los atributos deben ayudarnos a describir las clases aceptadas. No en vano, son los que dan sentido al objeto dentro del contexto del problema. Esto último es muy importante, ya que el contexto del problema puede hacer variar los atributos de un objeto. Por ejemplo, si imaginamos un sistema de gestión de una librería donde se está modelando la clase libro. Así, el objeto libro de librería debería contener atributos como precio de distribuidor, precio sin IVA o precio con IVA, mientras que esos atributos no serían necesarios en la gestión de préstamos de biblioteca, donde en cambio podríamos tener atributos como la asignatura y su disponibilidad.

Los atributos son las propiedades que definen al objeto. Al objeto podemos acceder a los atributos, ya sea para consultarlo o modificarlo mediante un punto. ejemplo: libro.precio

Rodríguez, R., Sosa, E., & Prieto, Á. (2004). Programación orientada a objetos

métodos

Los métodos son una colección de instrucciones de programa que serán ejecutadas por el objeto que permite recibir un mensaje con el mismo nombre del método invocado. Los métodos son implantaciones de las operaciones relevantes para una clase de objetos. Los métodos son invocados en respuesta a los mensajes.

los métodos son las acciones que se pueden realizar

Rodríguez, R., Sosa, E., & Prieto, Á. (2004). Programación orientada a objetos

instanciar

Los valores para las variables son proporcionados por cada instancia de la clase. De ésta manera, después de ser creada la clase, se deberá instanciar (creando una instancia) antes de que pueda ser utilizada.

Cuando se crea una instancia de una clase, se crea un objeto del tipo y el sistema operativo asignará memoria para las variables declaradas en la clase. Esto para darle valores a los atributos y para hacerlo se usa el método 'constructor()'.

El nombre del método constructor depende del lenguaje de programación. por ejemplo en python es init y en java es el mismo nombre de la clase

Rosiles, M. A. C. (2023). Programación orientada a objetos.

Abstracción:

cuando hacemos una clase debemos decidir cuáles atributos y métodos deben de tener depende lo que necesites

biblioteca de música

pagina de musica

canción	canción
atributos nombre artista duración género	atributos nombre artista precio
métodos escuchar() constructor()	métodos comprar() constructor()

encapsulamiento

Uno de los pilares básicos de la POO es la encapsulación de los datos. Según los principios de este paradigma de programación, el acceso a los datos de una clase debe realizarse de forma controlada, protegiéndolos de accesos no deseados. Cuando se desarrolla una aplicación, a veces es necesario ocultar los tipos de datos usados para que el usuario permanezca independiente de los detalles de los mismos. De esta manera, el usuario no es sensible a los cambios que se puedan producir en los tipos de datos elegidos dentro de una clase.

La encapsulación es el mecanismo que enlaza el código y los datos, a la vez que los asegura frente a accesos no deseados. La principal razón del uso de la encapsulación es evitar el acceso directo a atributos de una clase desde fuera de la propia clase. Se busca, principalmente, que el acceso a estos atributos se realice siempre mediante funciones miembro de la propia clase. El acceso a ciertos elementos de datos se puede controlar de forma estricta considerándolos privados.

puedes crear métodos de acceso y de modificación de los atributos, dentro de estos métodos podemos tener control sobre ellos. estos métodos se suelen llamar get para acceder y set para modificar

Rodríguez, R., Sosa, E., & Prieto, Á. (2004). Programación orientada a objetos.

herencia

En cuanto a la herencia, se basa en crear una nueva clase que especializa una clase existente, añadiéndole nuevo código sin modificar dicha clase base. Una de las ventajas de la herencia y la composición es que posibilitan el desarrollo incremental permitiendo introducir nuevo código sin causar errores en el código existente. Si aparecen errores, éstos estarán aislados dentro del nuevo código, que es mucho más fácil y rápido de leer que si se hubiese modificado

subclase	superclase/clase padre	subclase
guerrero	personaje	mag0
atributos	atributos	atributos

nombre fuerza inteligencia defensa vida posición contador	nombre fuerza inteligencia defensa vida posición contador	nombre fuerza inteligencia defensa vida posición contador
métodos atacar() moverse() turno() constructor() GetNombre() SetConstructor()	métodos atacar() moverse() turno() constructor() GetNombre() SetConstructor()	métodos atacar() moverse() turno() constructor() GetNombre() SetConstructor()
métodos y atributos agregados espada armadura habilidad() constructor()		métodos y atributos agregados libro magia() canstructor()

Rodríguez, R., Sosa, E., & Prieto, Á. (2004). Programación orientada a objetos.

poliformismo

Permite mejorar la organización del código y simplificar la programación. Además, aumenta las posibilidades de extensión y evolución de los programas. Está directamente relacionado con las jerarquías de clase. Básicamente, nos va a permitir que unos objetos tomen el comportamiento de objetos que se encuentran más abajo en la jerarquía, aumentando enormemente la expresividad del lenguaje. En definitiva, se trata de la posibilidad de que la identificación del tipo de un objeto se haga en tiempo de ejecución en vez de en tiempo de compilación.

permite a un método ser diferente según que clase lo este usando, es decir puede tener muchas formas de usarse

En el ejemplo de los personajes. Las tres clases tienen el método atacar que realizan la misma acción, pero después de la herencia podemos volver a definir los métodos heredados. por ejemplo: el método 'atacar'

se usa para atacar con las manos pero en la clase guerrero queremos que incremente el daño según el arma que use, por ejemplo: fuerza*espada.

si llamamos al método atacar para calcular cuánto daño ha realizado, según que clase reciba lo calculara de distintas formas

fuerza= 10

espada = 5

personaje jugador = newGuerrero():

interger daño = jugador.atacar();

interger daño = 50

Rodríguez, R., Sosa, E., & Prieto, Á. (2004). Programación orientada a objetos.