

Taller sobre programación orientada a objetos

Sebastian Arevalo matiz

Matallana

Universidad Uniminuto

Ingeniería de sistemas

Zipaquirá

2025

Tabla de contenido

| | |
|---|---|
| Tabla de contenido | 2 |
| Introducción | 3 |
| Objetivos | 4 |
| Qué es la programación orientada a objetos..... | 5 |
| Qué es un objeto | 5 |
| Diagramas de clases..... | 5 |
| La estructura de un diagrama de clases..... | 5 |
| Como se relacionan entre ellos | 5 |
| Mejores Herramientas de Diagrama UML en Línea Gratuitas..... | 5 |
| Cuales son los tipos de métodos..... | 6 |
| Cuáles son los principios de la programación orientada a objetos..... | 7 |
| Qué son las clases abstractas | 7 |
| Qué son las interfaces | 7 |
| Conclusiones..... | 8 |
| Bibliografías..... | 9 |

Introducción

La programación orientada a objetos o mejor llamada como POO es un modelo de programación que organiza el software en torno a objetos, en lugar de funciones y lógica facilitando la reutilización del código y su mantenimiento. Desde su surgimiento en el año de 1960, la Poo ha evolucionado constantemente hasta convertirse en uno de los enfoques mas efectivos para la creación de software.

La POO se basa en la idea de que los sistemas informáticos deben estar compuestos por objetos que interactúan entre si. Esto permite una mayor abstracción, modularidad y reutilización de código lo que facilita la creación de sistemas mas complejos y mantenibles.

En este trabajo vamos a ver varios conceptos acerca de la Poo como diagrama de clases ,clases ,objetos, relación, métodos, clases abstractas, interfaces y principios como la herencia, polimorfismo, encapsulamiento y abstracción ;en donde todos estos conceptos son muy fundamentales para la Poo y que debemos tenerlos en cuenta al momento de hablar sobre lo mencionado,

Objetivos

Objetivo general:

Analizar los conceptos fundamentales de la programación orientada a objetos.

Objetivos específicos:

- Identificar la importancia de la 'programación orientada a objetos para la organización de código.
- Reconocer las bases para el uso de la programación orientada a objetos.

1.¿Qué es la programación orientada a objetos?

Es un paradigma de programación que define los programas en términos de “clases de objetos”, objetos que son entidades que combinan estado (propiedades o datos), comportamiento (procedimientos o métodos) e identidad (propiedad del objeto que lo diferencia del resto).

2.¿Qué es un objeto?

Los objetos en programación se usan para modelar objetos o entidades del mundo real (el objeto hijo, madre, o farmacéutica, por ejemplo).

Un objeto es, por tanto, la representación en un programa de un concepto, y contiene toda la información necesaria para abstraerlo: datos que describen sus atributos y operaciones que pueden realizarse sobre los mismos.

3.Diagramas de clases

los diagramas de clases se dibujan en UML (lenguaje de modelado unificado) que ilustran los atributos, relaciones y operaciones entre objetos. Los diagramas de clases son la base del software y son diagramas de estructura estática que funcionan según el principio de orientación a objetos. Ayudan en el modelado de datos, el modelado conceptual y la traducción de los modelos a códigos de programación.

3.1La estructura de un diagrama de clases

-Nombre: La primera fila en una figura de clase.

-Atributos: La segunda fila en una figura de clase. Cada atributo de una clase está ubicado en una línea separada.

-Métodos: La tercera fila en una figura de clase. También conocidos como "operaciones", los métodos se organizan en un formato de lista donde cada operación posee su propia línea.

3.2Como se relacionan entre ellos:

En un diagrama de clases, todas las clases están vinculadas entre sí mediante relaciones adecuadas. Estos enlaces ayudan al usuario a comprender a fondo la conexión entre diferentes entidades

3.3Mejores Herramientas de Diagrama UML en Línea Gratuitas

-GitMind

-Visual Paradigm

-Violet UML editor

-Software Ideas Modeler

-UMLet 14.3

-ConceptDraw Diagram

-Visio

-Umbrello

4.¿Cuales son los tipos de métodos?

En Java, podemos encontrar diferentes tipos de métodos según su estructura y funcionalidad:

-Métodos sin retorno: Estos métodos son aquellos que no devuelven ningún valor. Se utilizan principalmente para ejecutar un conjunto de instrucciones o modificar el estado de un objeto.

```
public void saludar() {  
    System.out.println("¡Hola, mundo!");  
}
```

- Métodos con retorno: Estos métodos devuelven un valor después de su ejecución. Este valor puede ser de cualquier tipo de dato válido en Java, como enteros, cadenas de texto, booleanos, entre otros. Por ejemplo, el resultado de una operación aritmética.

```
public int sumar(int a, int b) {  
    return a + b;  
}
```

-Métodos estáticos: Los métodos estáticos están asociados a la clase en lugar de una instancia de la clase. Estos métodos se pueden invocar directamente sin la necesidad de crear un objeto.

```
public static void mostrarMensaje() {  
    System.out.println("¡Hola, soy un ejemplo de método estático!");  
}
```

Declaración de métodos

5.¿Cuáles son los principios de la programación orientada a objetos?

-Abstracción: Este principio consiste en simplificar la complejidad del mundo real modelando solo los aspectos relevantes para un propósito particular, mientras se ocultan los detalles innecesarios. La abstracción permite a los desarrolladores concentrarse en lo que es importante para la aplicación, sin necesidad de entender cada detalle del comportamiento o estado interno de los objetos. Esto reduce la complejidad y mejora la comprensibilidad del código.

-Encapsulamiento: El encapsulamiento es el concepto de ocultar los detalles internos del funcionamiento de un objeto, protegiendo su estado interno de cambios no autorizados y exponiendo solo una interfaz pública para la interacción. Esto se logra mediante el uso de métodos que actúan como los únicos puntos de acceso para los atributos del objeto. El encapsulamiento no solo protege la integridad de los datos, sino que también reduce las dependencias entre diferentes partes del código, facilitando su modificación y mantenimiento.

-Herencia: La herencia permite que una clase derive o herede propiedades y comportamientos de otra clase, conocida como su clase padre. Esto facilita la reutilización del código y la extensión de las funcionalidades existentes sin tener que reescribir mucho código. Además, la herencia promueve la creación de una jerarquía de clases que puede reflejar relaciones naturales entre objetos más amplios y específicos, mejorando la organización del código.

-Polimorfismo: El polimorfismo se refiere a la capacidad de un objeto para tomar varias formas y comportarse de manera diferente en diferentes contextos, usando la misma interfaz. Esto se logra mediante el uso de la herencia y la sobrescritura de métodos, permitiendo que un mismo método tenga diferentes implementaciones en distintas clases. El polimorfismo no solo hace el software más flexible y adaptable a nuevas situaciones, sino que también permite tratar objetos de diferentes clases de manera uniforme.

6.¿Qué son las clases abstractas?

Pueden proporcionar alguna implementación básica y dejar otros métodos completamente abstractos para que sean implementados por sus clases derivadas.

7.¿Qué son las interfaces?

Una recopilación de firmas de operaciones o de definiciones de atributo que define un conjunto uniforme de comportamientos. Las interfaces son similares a una clase, excepto por que una clase puede tener una instancia de su tipo, y una interfaz debe poseer, como mínimo, una clase para implementarla.

Una recopilación de firmas de operaciones o de definiciones de atributo que define un conjunto uniforme de comportamientos. Las interfaces son similares a una clase, excepto por que una clase puede tener una instancia de su tipo, y una interfaz debe poseer, como mínimo, una clase para implementarla

Conclusiones

La programación orientada a objetos (POO) es un paradigma de programación que ha revolucionado la forma en que se diseñan y desarrollan sistemas informáticos. La POO se basa en la idea de que los sistemas informáticos deben estar compuestos por objetos que interactúan entre sí.

Los diagramas de clases son una herramienta fundamental para la POO, ya que permiten visualizar los atributos, relaciones y operaciones entre objetos. Los diagramas de clases son la base del software y son diagramas de estructura estática que funcionan según el principio de orientación a objetos.

La POO se basa en principios como la abstracción, el encapsulamiento, la herencia y el polimorfismo. Estos principios permiten la creación de software organizado, adaptable y mantenible.

Las clases abstractas y las interfaces son herramientas importantes en la POO, ya que permiten la creación de software flexible y extensible. Las clases abstractas pueden proporcionar implementación básica y dejar otros métodos completamente abstractos para que sean implementados por sus clases derivadas. Las interfaces son una recopilación de firmas de operaciones o de definiciones de atributo que define un conjunto uniforme de comportamientos.

En resumen, la POO es un paradigma de

<https://openwebinars.net/blog/introduccion-a-java-metodos-parametros-y-argumentos/> programación que facilita el desarrollo de software organizado, adaptable y mantenible. La POO se basa en principios como la abstracción, el encapsulamiento, la herencia y el polimorfismo, y utiliza herramientas como los diagramas de clases, las clases abstractas y las interfaces para crear software flexible y extensible.

Bibliografías

-Rivas García(s.f).Programación orientada a objetos.

<https://www.conalepveracruz.edu.mx/iniciobackup/wp-content/uploads/2021/03/Programación-orientada-a-objetos-MÓDULO-PROFESIONAL.pdf>

-Relación es en diagramas de clases UML(s.f) .Edrawsoft.

https://www.edrawsoft.com/es/article/class-diagram-relationships.html?srsId=AfmBOop_J1qrFv0ncTk4IV8EitzDdIXzmLzyKqz_2rmc-5YTgzgwlMQ

-Valencia(9 de mayo de 2024).Qué es la Programación Orientada a Objetos (POO) y cuáles son sus principios fundamentales

<https://coderslink.com/talento/blog/que-es-la-programacion-orientada-a-objetos-poo-y-cuales-son-sus-principios-fundamentales/>

-Ale Green(9 de agosto de 2021). Las 8 Mejores Herramientas Gratuitas de Diagramas UML en Línea en 2025,gitmind.

<https://gitmind.com/es/herramienta-gratis-diagrama-uml.html>

- Tutorial de diagrama de clases UML(s.f). Lucidchart.

<https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml>

-Barragán(24 de octubre de 2023).Introducción a java: Métodos, parámetros y argumentos,openwebinars.

<https://openwebinars.net/blog/introduccion-a-java-metodos-parametros-y-argumentos/>