

Tabla de contenido

Introducción.....	2
¿Qué es la programación orientada a objetos?	2
¿Por qué se utiliza la programación orientada a objetos?	2
¿Cómo funciona la programación orientada a objetos?	2
¿Cuándo se usa la programación orientada a objetos?	3
¿Dónde se usa la programación orientada a objetos?	3
¿Cuáles son las ventajas de la programación orientada a objetos?	3
¿Cuáles son las desventajas de la programación orientada a objetos?	4
¿Qué es una clase y un objeto en la programación orientada a objetos?	4

Introducción F. Behn

Este texto presenta una introducción a la programación orientada a objetos, un enfoque de programación que se centra en crear objetos y hacerlos interactuar para resolver problemas. Se abordan varios aspectos de la programación orientada a objetos, como su funcionamiento, cuándo se utiliza, sus ventajas y desventajas, así como las clases y objetos que conforman este paradigma.

Asimismo, se incluyen ejemplos de cómo la programación orientada a objetos se aplica para modelar entidades en un programa. Esta documentación resulta útil tanto para personas que se inician en la programación orientada a objetos como para aquellas que buscan repasar los conceptos fundamentales de este paradigma.

¿Qué es la programación orientada a objetos? F. Behn

La programación orientada a objetos es un paradigma de programación que se enfoca en la creación de objetos y su interacción para resolver problemas.

¿Por qué se utiliza la programación orientada a objetos?

La programación orientada a objetos se utiliza porque permite un mayor modularidad, reutilización de código, flexibilidad y organización en el desarrollo de software. F. Behn

¿Cómo funciona la programación orientada a objetos?

La programación orientada a objetos es un paradigma de programación que se basa en la idea de organizar el código en torno a objetos. Estos objetos son entidades que tienen propiedades (atributos) y comportamientos (métodos) asociados. F. Jordán

La programación orientada a objetos contiene 4 pilares:

- **Abstracción:** Consiste en considerar aisladamente uno de los caracteres de un objeto dado, es concebir una cualidad sin tener en cuenta la substancia o viceversa, es considerar la substancia dejando a un lado la cualidad. No es necesario darles a los objetos un parecido total a los objetos de la vida, por lo que se puede abstraer mucha de la información del mundo real para posteriormente crear los objetos en programación orientada a objetos.
- **Encapsulación:** Al encapsular algo, esto se vuelve privado, bloqueando el contenido al exterior de su clase, de esto es capaz un objeto. El modelo protegido es un poco menos restrictivo y hace que un miembro de una clase también esté disponible para las subclases.

- Herencia: La herencia es crear clases nuevas utilizando los comportamientos y estados de la clase ya existente. Si se quiere añadir funcionalidades a una clase, pero sin tocar la superclase, entonces se puede crear una subclase que herede todo. La consecuencia del uso de la herencia es que las subclases tienen la misma interfaz que su clase padre. No se puede esconder un método en una subclase si se declaró en la superclase.
- Polimorfismo: El polimorfismo permite tratar objetos de diferentes clases de manera uniforme. Aunque los objetos pertenezcan a clases diferentes, se pueden invocar los mismos métodos en ellos sin tener que conocer la clase específica del objeto.

¿Cuándo se usa la programación orientada a objetos? L. Baz

La programación orientada a objetos se utiliza en aquellos casos en los que se necesita un enfoque estructurado. Este paradigma de programación se basa en la creación de objetos que interactúan entre sí para llevar a cabo una función.

¿Dónde se usa la programación orientada a objetos? L. Baz

La programación orientada a objetos se usa para modelar entidades concretas en el programa. Por ejemplo, se puede crear una clase "Gato" con atributos como nombre, sexo, edad, peso, color, comida favorita, etc. Cada gato es una instancia de la clase "Gato" y tiene valores únicos de sus atributos.

¿Cuáles son las ventajas de la programación orientada a objetos? N. Núñez

- En la programación orientada a objetos se pueden desarrollar programas con módulos que se comunican entre sí (en lugar de tener que desarrollarlos desde cero), lo que ahorra tiempo y esfuerzo.
- La programación orientada a objetos es una tecnología que ofrece mayor productividad y menor costo de mantenimiento.
- La programación orientada a objetos se puede actualizar de sistema pequeño a grande fácilmente.
- En la programación orientada a objetos es fácil dividir el trabajo.
- En la programación orientada a objetos es fácil ocultar datos, lo que ayuda a crear programas más seguros.
- Gracias a la herencia, se puede reducir el código redundante y sacarles más partido a las clases existentes.

¿Cuáles son las desventajas de la programación orientada a objetos? N. Núñez

- Los programas desarrollados con la programación orientada a objetos pueden ser más pesados y lentos.
- La programación orientada a objetos no puede ser aplicado en todas partes, ya que no es un lenguaje universal.
- Solo debe utilizarse cuando es requerido, no puede ser usado para solucionar cualquier tipo de problemas.
- La programación orientada a objetos es más complicada que el promedio, por lo que se requieren mayores conocimientos para emplearlo.

¿Qué es una clase y un objeto en la programación orientada a objetos? F. Jordán

Una clase es una plantilla o plano utilizado para crear objetos, qué contiene definidas las características y comportamientos de un montón de objetos. Representa un concepto abstracto, encapsula datos y las operaciones que se aplican a estos datos.

Un objeto es una instancia o ejemplar concreto de una clase. Se crea a partir de la clase y tiene su propia identidad y estado único. Un objeto tiene atributos (variables) que almacenan su estado y métodos (funciones) que definen su comportamiento. Se pueden crear infinitos objetos a partir de una misma clase, cada uno puede tener sus valores y acciones específicos.

O sea que una clase define propiedades y comportamientos que comparten los objetos, y los objetos son las instancias individuales que se crean a partir de una clase y representan entidades concretas en el programa. La programación orientada a objetos permite modelar y organizar el código de una manera más estructurada y modular, lo que facilita la reutilización y la comprensión del código.