**1. Investigar las principales características de la Programación Orientada a Objetos.**

La idea principal de la POO es que los objetos interactúan entre sí, y estos están organizados en clases, dentro de las principales características están:

* **Abstracción:**

Captura los detalles más importantes de los objetos.

* **Encapsulamiento:**

Consiste en agrupar los elementos de un objeto en un entorno bien definido.

* **Polimorfismo:**

Permite implementar múltiples formas de un mismo método, es decir, sobrecargar un objeto el cual consiste en que varios métodos tengan el mismo nombre pero retornen diferentes resultados o realicen diferentes acciones.

* **Herencia:**

Es el proceso mediante el cual un objeto adquiere las propiedades de otro objeto, es decir, que se pueden compartir los métodos y atributos entre las clases y subclases

**2. ¿En qué se diferencia la Programación Orientada a Objetos y la Programación Estructurada?**

En el uso de los objetos, clases y métodos, los cuales permiten el encapsulamiento, la herencia y la interacción entre sí, permitiendo la reutilización de código; detalles que no se pueden realizar con la programación estructurada.

**3. Realice una investigación referente a los conceptos de herencia, encapsulamiento, polimorfismo. Definir sus características y ventajas en el desarrollo de Software.**

**La herencia** Es el proceso mediante el cual un objeto adquiere las propiedades de otro objeto, es decir, que se pueden compartir los métodos y atributos entre las clases y subclases.

Existen los siguientes tipos de herencia:

* **Herencia única:** donde las subclases heredan las características de solo una superclase.
* **Herencia Multinivel:** una clase derivada heredará una clase base y, además, la clase derivada también actuará como la clase base de otra clase.
* **Herencia Jerárquica:** una clase sirve como una superclase (clase base) para más de una subclase.
* **Herencia Múltiple (a través de interfaces):** una clase puede tener más de una superclase y heredar características de todas las clases principales. Pero Java no admite herencia múltiple con clases, así que para lograrlo tenemos que usar Interfaces.
* **Herencia Híbrida (a través de Interfaces):** Es una mezcla de dos o más tipos de herencia anteriores. Como Java no admite herencia múltiple con clases, la herencia híbrida tampoco es posible con clases, pero como en el ejemplo anterior, podemos lograr el mismo resultado a través de Interfaces.

**El Encapsulamiento**, consiste en agrupar los elementos de un objeto en un entorno con limites bien definido, evitando el acceso a datos por cualquier otro medio distinto a los especificados.

Existen tres niveles de acceso para el encapsulamiento, los cuales son:

* **Público (Public):** Todos pueden acceder a los datos o métodos de una clase que se definen con este nivel, este es el nivel más bajo, esto es lo que tu quieres que la parte externa vea.
* **Protegido (Protected):** Podemos decir que estás no son de acceso público, solamente son accesibles dentro de su clase y por subclases.
* **Privado (Private):** En este nivel se puede declarar miembros accesibles sólo para la propia clase.

**Polimorfismo:**

Permite implementar múltiples formas de un mismo método, es decir, sobrecargar un objeto el cual consiste en que varios métodos tengan el mismo nombre pero retornen diferentes resultados o realicen diferentes acciones.

Por lo general diremos que existen 3 tipos de polimorfismo:

* **Sobrecarga:** El más conocido y se aplica cuando existen funciones con el mismo nombre en clases que son completamente independientes una de la otra.
* **Paramétrico:** Existen funciones con el mismo nombre pero se usan diferentes parámetros (nombre o tipo). Se selecciona el método dependiendo del tipo de datos que se envíe.
* **Inclusión:** Es cuando se puede llamar a un método sin tener que conocer su tipo, así no se toma en cuenta los detalles de las clases especializadas, utilizando una interfaz común.

**4. ¿Qué beneficios principales nos da la Programación Orientada a Objetos?**

* Aplicaciones más seguras, robustas y confiables.
* Permite la reutilización del código, lo cual permite desarrollos más rápidos y eficientes.
* Fácil relación entre sí de los objetos.
* Sobrecarga de los objetos.

FORO MODULO 3

**¿Que representan los objetos en Java?**

Representan su principal enfoque, ya que este lenguaje funciona en base a los objetos bien definidos y su relación entre sí, ya sea que estos sean desarrollados o propias del lenguaje.

**¿Qué tanto puede optimizar el código con la implementación de los patrones de diseño?**

De gran manera, ya que por medio dela interacción o la herencia de las clases permite el ahorro de código y permite la optimización del mismo, también fomenta la estandarización del código, punto muy importante para una mejor compresión de los desarrolladores.

**¿Qué fase debe tener mayor importancia en el desarrollo de aplicaciones:**

1. **Análisis:** La fase más importante de un proyecto, ya que nos permite recopilar toda la información necesaria al igual comprender y entender el problema o el proceso requerido para su solución o automatización. Un buen análisis no solo nos facilitara el desarrollo del proyecto, también nos evitara gastos adicionales.
2. **Diseño:** Ocupa la segunda posición de las acciones a realizar en el proyecto, siempre y cuando se realice un buen análisis, el diseño fluirá con más facilidad, acá podemos definir los objetos necesarios y la relación entre sí para el desarrollo del proyecto
3. **Implementación:** Siempre y cuando el análisis y el diseño se realicen de la manera correcta la fácil final del proyecto que es la implementación seria de lo más sencilla y sin complicaciones.

Detallar su respuesta y colocar esas tres fases que consideren en orden de mayor a menor relevancia para el éxito en los proyectos.

Vencimiento 17 de Junio, 2020, 11:59 PM