Teleinformática

ADSO

Carlos Eduardo Pastuso Bravo

Profesora:

Jennifer fajardo

SENA

Diseño y Programación Orientada a Objetos

Bogotá-Fontibón

Noviembre / 2022

Los 4 pilares de la programación (Encapsulamiento, Abstracción, Herencia, y polimorfismo)

1. Explique puntualmente al concepto

Encapsulamiento:

es la forma en la que podemos definir como se visualiza la información o como será oculta, esto para la seguridad de la misma, un atributo principalmente aparece como un campo público, pero podemos volverlo Privado o en su defecto para otras cosas dejarlo protegido.

Nos permite controlar quien puede ver y utilizar los distintos módulos internos de nuestro sistema. En términos de clases, con el encapsulamiento definimos el acceso a los miembros de la clase.

Herencia:

 Pero se define como la capacidad que tiene una clase padre de heredad sus atributos y métodos a una clase hija, y aunque hereda los atributos la clase hija tiene atributos propios.

Es una relación especial entre dos clases, la clase base y la clase derivada, en donde la clase derivada obtiene la habilidad de utilizar ciertas propiedades y funcionalidades de la clase base, incluso pudiendo sustituir funcionalidad de la clase base. La idea es que la clase derivada “hereda” algunas de las características de la clase base.

Polimorfismo:

 definir clases diferentes que tienen métodos o atributos denominados de forma idéntica, pero se comportan diferente.

tener una función la cual reciba un parámetro, como una clase base, y podemos pasarle a ese método objetos que sean instancias de las clases derivadas de dicha clase base. Lo mismo ocurre si el método recibe como parámetro una interfaz. Podemos pasarle a dicho método cualquier clase que implemente dicha interfaz.

Abstracción:

es la capacidad de obtener y aislar toda la información y cualidades de un objeto que no nos parezcan relevantes, para poder encapsularlos. Para ello separamos "mentalmente" los objetos y nos centramos en su comportamiento fundamental.

Cuando hacemos una abstracción, queremos omitir detalles que no son necesarios para nosotros, y queremos solamente mostrar lo que sí es relevante.

Ejemplos

1. Encapsulamiento: Imagina un método de búsqueda de contactos. El usuario puede buscar por diferentes criterios, cada nuevo criterio de búsqueda se convierte en un nuevo parámetro del método, Para resolver este problema podemos crear una nueva abstracción que se llame Criterios De Búsqueda y en ella encapsular los parámetros de búsqueda. Esto nos permite pasar de varios parámetros a uno solo.
2. Herencia: vamos a simular el comportamiento que tendrían los diferentes integrantes de la selección de futbol; tanto los Futbolistas como el cuerpo técnico (Entrenadores, Masajistas, etc…). Para simular este comportamiento vamos a definir tres clases que van a representaran a objetos Futbolista, Entrenador y Masajista, vemos que en las tres clases tenemos atributos y métodos que son iguales ya que los tres tienen los atributo, ya que las tres clases tienen métodos y atributos comunes, de ahí y como veremos enseguida, decimos que la herencia consiste en "***sacar factor común***".
3. Polimorfismo: Gato y Perro, que heredan de la superclase Animal. La clase Animal tiene el método abstracto, que se implementa de forma distinta en cada una de las subclases (gatos y perros suenan de forma distinta). Entonces, un tercer objeto puede enviar el mensaje de hacer sonido a un grupo de objetos Gato y Perro por medio de una variable de referencia de clase Animal, haciendo así un uso polimórfico de dichos objetos respecto del mensaje mover.
4. Abstracción: En una consulta médica, el doctor va desechando la información innecesaria que dice el paciente y acumula mentalmente solo aquella que le aporta datos de la enfermedad del paciente

Infografía

Los 4 pilares de la programación

Encapsulamiento

Permite reunir a todos los elementos pertenecientes a una misma entidad

Herencia

Evitar que tener que escribir el mismo código una y otra vez, estamos atribuyéndoles las características generales de la primera

Abstracción

Dividen los objetos que se agrupan para formar cosas más complejas, básicamente separa los elementos.

Polimorfismo

Pueden contener objetos de diferentes tipos, producirá el comportamiento correcto para el tipo real