**Programación Orientada a Objetos**

La programación orientada a Objetos (POO) es una forma de programación que utiliza objetos, ligados mediante mensajes, para la solución de problemas. Puede considerarse como una extensión natural de la programación estructurada en un intento de potenciar los conceptos de modularidad y reutilización de código.

**Mecanismos básicos de la POO**

Los mecanismos de la programación orientada a objetos son: objetos, mensajes, métodos y clases.

* **Objetos:**

Un programa orientado a objetos se compone de solamente de objetos. Un objeto es una encapsulación genérica de datos y de los procedimientos para manipularlos. Dichos de otra forma un objeto es una entidad que tiene unos atributos particulares, las propiedades, y unas formas de operar sobre ellos, los métodos. Por lo tanto, un objeto contiene, por una parte, operaciones que definen sus comportamiento, y por otra, variable manipuladas por esas operaciones que definen su estado.

Por ejemplo: una ventana de Windows es un objeto. El color de fondo de la ventana, el ancho, el alto, etc, son propiedades. Las rutinas, lógicamente transparentes al usuario, que permiten maximizar la ventana, minimizar, etc. Son métodos.

* **Mensajes:**

Cuando se ejecuta un programa orientado a objetos, los objetos están recibiendo. Interpretando y respondiendo a mensajes de otros objetos. Esto marca una clara diferencia con respeto a los elementos de datos pasivos de los sistemas tradicionales. Por ejemplo, en Visual Basic un mensaje esta asociado con un procedimiento, de tal forma que cuando un objeto recibe un mensaje la respuesta a ese mensaje es ejecutar el procedimiento asociado. Este procedimiento recibe el nombre de Método.

* **Método:**

Un método se implementa en una clase de objeto y determina como tiene que actuar el objeto cuando recibe un mensaje. En adición, las propiedades permitirán almacenar información para dichos objetos. Un método puede también enviar mensajes a otros objetos solicitando una acción o información.

Método

Objetos

Mensajes

Datos

La estructura más interna de un objeto

esta oculta para otros usuarios y la única

conexión que tiene con el exterior son los

mensajes. Los datos que están dentro

de un objeto solamente pueden ser manipulados

 por los métodos asociados al propio objeto.

La ejecución de un programa orientado a objeto realiza fundamentalmente tres cosas:

1. Crear los objetos necesarios.
2. Los mensajes enviados a unos y a otros objetos dan lugar a que se procese internamente la información.
3. Finalmente, cuando los objetos no son necesarios, son borrados, liberándose la memoria ocupada por los mimos.
* **Clases**

Una clase es un tipo de objetos definido por el usuario. Una clase equivale a una generalización de un tipo específico de objetos. Por ejemplo: una torta, el molde de torta es la clase y la torta el objetos.

Un objeto es una determinada clase se crea en el momento en que se declara una variable de dicha clases.

**CARACTERISTICAS DE LA POO**

Las características fundamentales de la POO son: Abstracción, encapsulamiento, herencia y polimorfismo.

* **Abstracción**

Por medio de la abstracción conseguimos no detenernos en los detalles concretos de las cosas que no interesen en cada momento, sino generalizar y centrarse en los aspectos que permitan tener una visión global del tema. Por ejemplo, el estudio de un ordenador podemos realizarlo a nivel de funcionamiento de sus circuitos electrónicos, en términos de corrientes, tensión, etc. o a nivel de transferencia entre registros, centrándose así el estudio en el flujo de información entre unidades que lo componen (memoria, unidad aritmética, unidad de control, registro, etc.), sin importarnos el comportamiento de los circuitos electrónicos que componen unidades.

* **Encapsulamientos**

Esta característica permite ver un objeto una caja negra en la que se ha metido de alguna manera toda la información relacionada con dicho objeto. Esto nos permitirá manipular los objetos cono unidades básicas, perteneciendo oculta su estructura interna.

La abstracción y el encapsulamiento están representadas por clase. La clase es una abstracción, porque en ella se definen las propiedades de un determinado conjunto de objetos con características comunes, y es una encapsulación, porque constituye una caja negra que encierra tanto los datos de que almacena como los métodos que permiten manipularlos.

* **Herencia**

La herencia es el mecanismo para compartir automáticamente métodos y datos entre clases y subclases de objetos. El usuario de Visual Basic no dispone de estas características.

* **Polimorfismo**

Esta característica permite implementar múltiples formas de un mismo método, dependiendo cada una de ellas de la clase sobre que se realice la implementación. Esto hace que se pueda acceder a una variedad de métodos distintos (todos con el mismo nombre) utilizando exactamente el mismo medio de acceso.