Universidad Nacional Autónoma de México

Colegio de Ciencias y Humanidades Plantel Sur.

Numero de lista: 2

Emilio Álvarez Quijano.

Grupo 601

Cibernética y Computación II.

Alberto Ávila

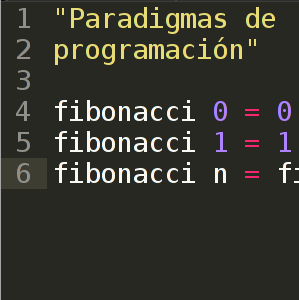
Actividad 1: Programación Orientada a Objetos.

Fecha de entrega 28 de abril de 2020



1. **¿Qué es un paradigma de programación?**

Un paradigma de programación es un estilo de desarrollo de programas. Es decir, un modelo para resolver problemas computacionales. Los lenguajes de programación, necesariamente, se encuadran en uno o varios paradigmas a la vez a partir del tipo de órdenes que permiten implementar, algo que tiene una relación directa con su sintaxis.



1. **¿Qué es la programación Orientada a Objetos (POO)?**

La programación orientada a objetos (POO, u OOP según sus siglas en inglés) es un paradigma de programación que viene a innovar la forma de obtener resultados. Los objetos manipulan los datos de entrada para la obtención de datos de salida específicos, donde cada objeto ofrece una funcionalidad especial.

Muchos de los objetos prediseñados de los lenguajes de programación actuales permiten la agrupación en bibliotecas o librerías, sin embargo, muchos de estos lenguajes permiten al usuario la creación de sus propias bibliotecas.



1. **¿Cuáles son los componentes de la Programación Orientada a Objetos (POO)?**

**Atributo:**

Los atributos son las propiedades o estados de un elemento Clase u objeto, los atributos se declaran variables del elemento y ayudan a estructurar el objeto de la clase.

**Método:**

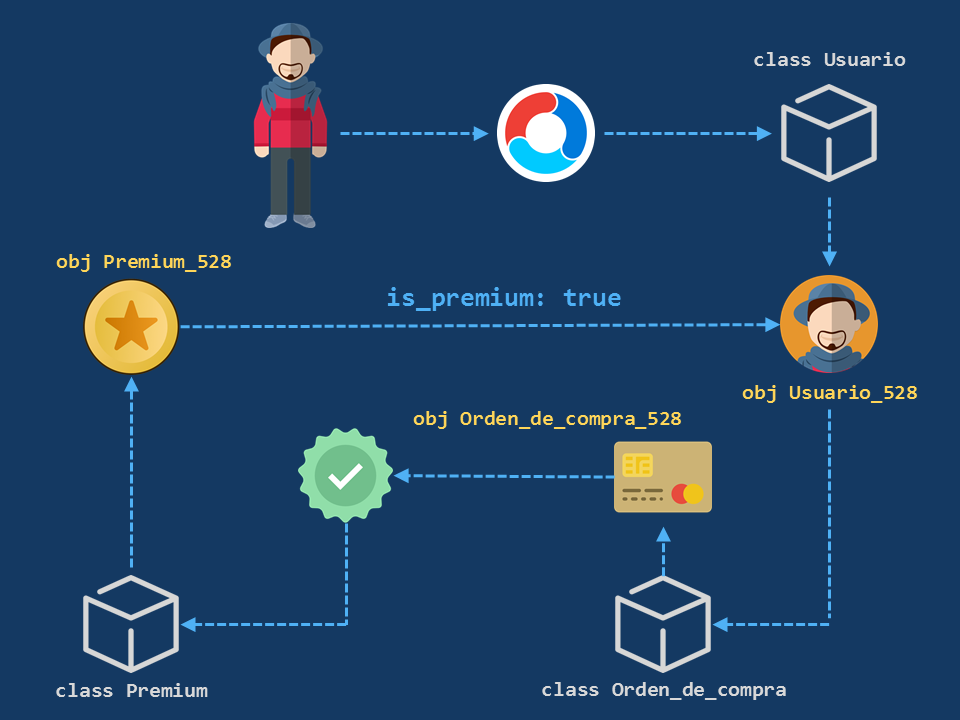
Los métodos son el conjunto de funciones que pueden tener los elementos de un POO clase u objeto, pueden ser funciones aritméticas, comparación, medición, etc. Estos métodos pueden ser privados o públicos.

**Clase:**

Una clase en POO es una plantilla para la creación de objetos, esta plantilla es un elemento genérico o básico que tiene las características generales, comportamientos, atributos del objeto que se quiera crear.

**Objeto e instancia**:

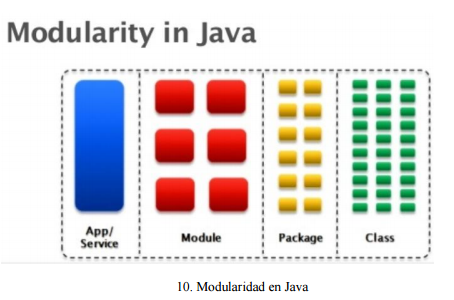
Es una unidad de programa que consta de atributos o propiedades y de funciones o métodos, esta unidad de programa se desarrolla a partir de una plantilla o clase, el desarrollo del objeto a partir de una clase (plantilla) se llama instancia**.**



1. **Menciona una breve descripción del concepto de modularidad**

La propiedad que permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes.

Estos módulos que se puedan compilar por separado, pero que tienen conexiones con otros módulos. Al igual que la encapsulación, los lenguajes soportan la Modularidad de diversas formas. La modularidad debe seguir los conceptos de acoplamiento y cohesión.



1. **¿Cuáles son las características de la POO?**

\*Abstracción

Es el proceso de capturar los detalles fundamentales de un objeto mientras se suprimen o ignoran detalles.

La abstracción proporciona un mecanismo crucial para permitir que las personas comprendan, se comuniquen y razonen sistemas complejos.

\*Encapsulamiento.

Es el proceso de ocultar todos los detalles de un objeto que no contribuyen a sus características esenciales.

La ventaja principal del encapsulamiento es que proporciona al programador libertad en la implementación de los detalles del sistema. La única restricción que tendrá el programador es mantener la interfaz abstracta que vean los usuarios externos.

\*Polimorfismo.

Es la propiedad que permite enviar el mismo mensaje a objetos de diferentes clases, de forma que cada uno de ellos responde a ese mismo mensaje de modo distinto dependiendo de su implementación.

El polimorfismo como sobrecarga en Java ocurre cuando una clase T tiene múltiples métodos con el mismo nombre, cada uno de ellos con una signatura (combinación de su nombre y el tipo y número de argumentos que pasan por los métodos) diferente

\*Herencia.

Es el proceso mediante el cual un objeto adquiere las propiedades de otro objeto.

Clases, Objetos y Mensajes

Clase: Es un prototipo o modelo que define los atributos y métodos comunes a todos los objetos de cierto tipo.

Objeto: Es una instancia (un ejemplar, un caso concreto) de una clase.

Mensaje: La interacción entre objetos se produce mediante mensajes. Los mensajes son llamados a métodos de un objeto en particular.

1. **¿Cuáles la definición de polimorfismo?**

El polimorfismo se refiere a la propiedad por la que es posible enviar mensajes sintácticamente iguales a objetos de tipos distintos. El único requisito que deben cumplir los objetos que se utilizan de manera polimórfica es saber responder al mensaje que se les envía.

La apariencia del código puede ser muy diferente dependiendo del lenguaje que se utilice, más allá de las obvias diferencias sintácticas.

Por ejemplo, en un lenguaje de programación que cuenta con un sistema de tipos dinámico (en los que las variables pueden contener datos de cualquier tipo u objetos de cualquier clase) como Smalltalk se requiere que los objetos que se utilizan de modo polimórfico sean parte de una jerarquía de clases.



1. **¿Qué es herencia?**

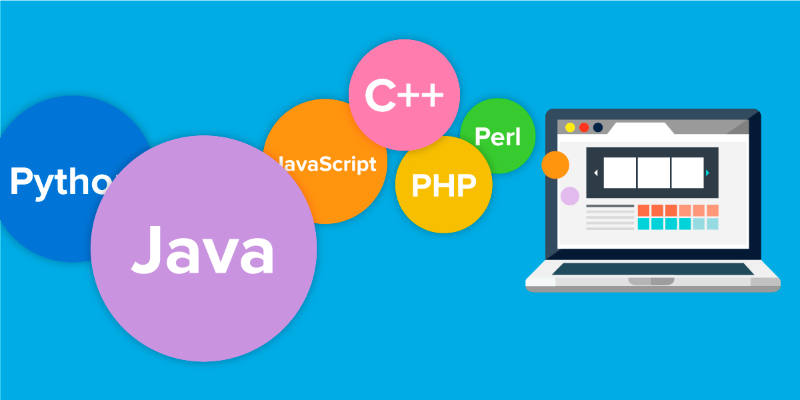
Es el mecanismo por el cual una clase permite heredar las características (atributos y métodos) de otra clase.

La herencia permite que se puedan definir nuevas clases basadas de unas ya existentes a fin de reutilizar el código, generando así una jerarquía de clases dentro de una aplicación. Si una clase deriva de otra, esta hereda sus atributos y métodos y puede añadir nuevos atributos, métodos o redefinir los heredados.



1. **¿Investiga algunos lenguajes de programación orientada a objetos?**

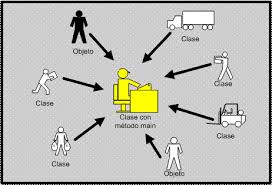
Actualmente existen una gran variedad de lenguajes de programación orientados a objetos entre los que podemos destacar: C++, Objective C (es el lenguaje principal de programación para Mac OS X , iOS), Java, Ruby, Visual Basic, Visual C Sharp, Perl, Smalltalk, PHP o Python.



1. **¿Qué es una clase en programación orientada a objetos?**

Una clase es una plantilla para la creación de objetos de datos según un modelo predefinido. Las clases se utilizan para representar entidades o conceptos, como los sustantivos en el lenguaje. Cada clase es un modelo que define un conjunto de variables -el estado, y métodos apropiados para operar con dichos datos -el comportamiento. Cada objeto creado a partir de la clase se denomina instancia de la clase.

Las clases de objetos son un pilar fundamental de la programación orientada a objetos. Permiten abstraer los datos y sus operaciones asociadas al modo de una caja negra. Los lenguajes de programación que soportan clases difieren sutilmente en su soporte para diversas características relacionadas con clases.

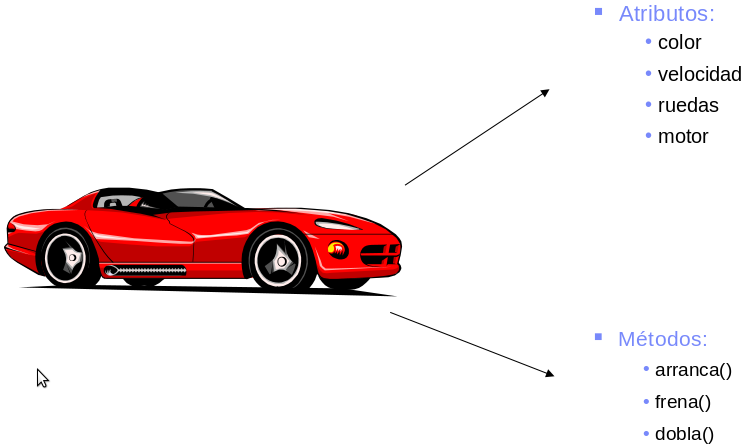


1. **¿Menciona algún ejemplo de objeto?**

Un auto es un objeto. También tiene un estado: Cantidad de puertas, color, tamaño, etc. y un comportamiento: acelerar, frenar, subir cambio, bajar cambio, girar izq., girar der., etc.

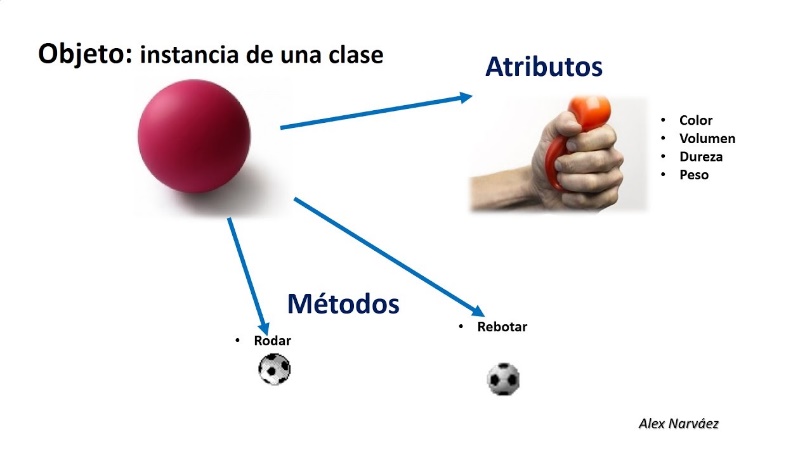
Entonces podemos definir a un objeto en POO, como un conjunto de datos y funciones relacionadas. A las funciones de los objetos, tales como acelerar en el caso del auto, de aquí en más las llamaremos métodos, a los datos los llamaremos atributos.

Los objetos en programación son modelados observando objetos del mundo real, por ejemplo, implementamos el objeto "perro" dentro de nuestro programa definiendo los atributos y métodos del objeto perro real.



1. **¿Qué es un método en programación orientada a objetos?**

Un método es una subrutina cuyo código es definido en una clase y puede pertenecer tanto a una clase, como es el caso de los métodos de clase o estáticos, como a un objeto, como es el caso de los métodos de instancia. Análogamente a los procedimientos en lenguajes imperativos, un método consiste generalmente de una serie de sentencias para llevar a cabo una acción, un juego de parámetros de entrada que regularán dicha acción o, posiblemente, un valor de salida (o valor de retorno) de algún tipo.



Referencias:

* Wikipedia. (s. f.). Programación Orientada a Objetos. Recuperado 27 de abril de 2020, de <https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos>
* Programacion.net. (s. f.). Programación Orientada a Objetos. Recuperado 27 de abril de 2020, de <https://programacion.net/articulo/programacion_orientada_a_objetos_279>