



Ciclo 2

Semana 1

Introducción a la Programación Orientada a Objetos (POO)

Lectura 2 - Elementos de un programa

Elementos de un programa



A continuación, se definirán brevemente los elementos de un programa, pero algunos de estos serán explicados de manera amplia más adelante cuando veamos el paradigma de programación orientada a objetos.

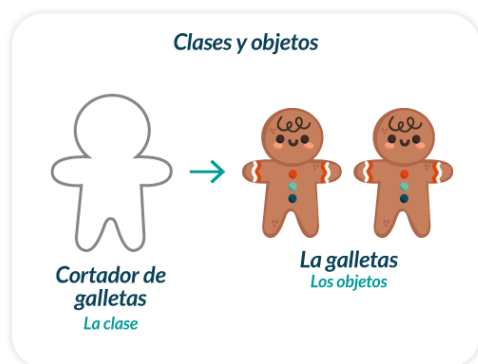


Algoritmos

Se pueden entender como una secuencia ordenada de instrucciones para realizar una actividad específica y que se ejecutaran para resolver un problema.

Cuando estos algoritmos ya son ejecutados por una computadora, estas instrucciones no responderán a un sentido común por tanto deben estar definidas y escritas sin ningún tipo de ambigüedad traducidas a un lenguaje de programación los cuales tiene reglas

definidas denominadas sintaxis del lenguaje.



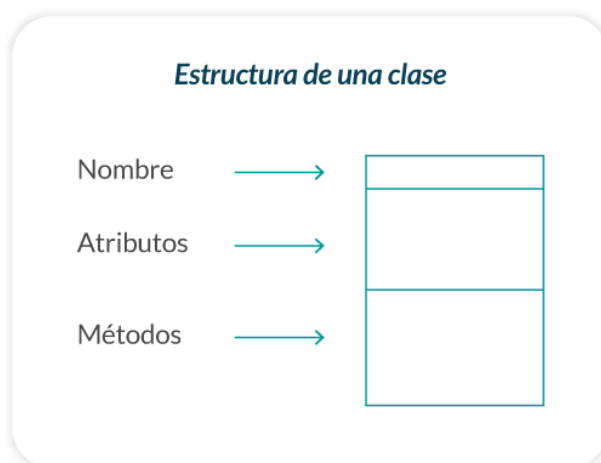
Clases y Objetos

Las clases son la representación de la estructura y comportamiento de un objeto y representan las entidades del mundo del problema. Puede entenderse como un patrón que define los atributos y métodos de un tipo particular de objetos o como un conjunto o agrupamiento de características (atributos) y acciones (métodos) que representan un elemento del mundo del problema, como un escenario general no particular.

Así pues, en un programa se encontrará una clase por cada entidad definida en el mundo del problema, pero, aunque es un patrón de una entidad, sobre estos pueden existir varias "instancias" para cada entidad. Por ejemplo, aunque una entidad definida es un automóvil, y estos tengan características comunes, se pueden encontrar diferentes tipos de autos.

Aquí debemos introducir el concepto de objeto, que es la base del paradigma de programación orientada a objetos y corresponde a una instancia de una clase y aunque la clase define unos atributos y este reúne las características de la clase, el objeto tiene sus propios valores para cada uno de estos atributos. Este conjunto de valores de los atributos se conoce como el estado del objeto.

En resumen, la clase define un tipo de elemento en el mundo y el objeto representa un elemento en particular y aunque todos los objetos de una clase dada son idénticos en estructuras y comportamientos son únicos, aunque tengan los mismos valores de sus atributos.





Una clase en la programación orientada a objetos es una estructura de datos que especifica un estado interno de un objeto llamado atributos, que tiene un comportamiento llamado métodos. Una manera fácil de entender que es una clase es haciendo una analogía de un patrón o molde, que especifica un escenario general definiendo cuales los atributos con la asignación de un nombre y un tipo de valor, pero sin indicar el valor de un atributo que tienen las instancias u objetos de esa clase.

Ejemplo práctico

Nombre	→	Reloj
Atributos	→	<ul style="list-style-type: none">• día: String• hora: String• modelo: String• numSerie: long
Métodos	→	<ul style="list-style-type: none">• getDia()• getHora()• incrementarDia()• incrementarHora()• limpiarPantalla()

Los métodos se deben definir en la clase y no se copian en cada objeto, así cuando se crea un objeto se determina a que clase pertenece y así este conoce los métodos que puede ejecutar. De esta manera los métodos resuelven los problemas de manera abstracta pero cada objeto es quien los aplica en un escenario particular para así dar solución a un problema concreto.

Podríamos entender el concepto de clase como un modelo, una plantilla o un esquema pues contiene todas las características comunes de un objeto, y de esta se pueden crear objetos que, aunque independientes tienen las mismas características.

La manera gráfica de representar una clase es con un rectángulo que se divide horizontalmente en tres secciones. En el compartimiento superior se coloca el nombre de la clase, en el medio los atributos colocando primero el nombre y separado con dos puntos el tipo de dato. En la última sección se listarán los métodos, definiendo primero un nombre de método y entre paréntesis sus parámetros separados por comas especificando su tipo, separado de dos puntos se coloca finalmente el tipo de retorno. Los métodos constructores no tienen tipo de retorno.



Semana 1

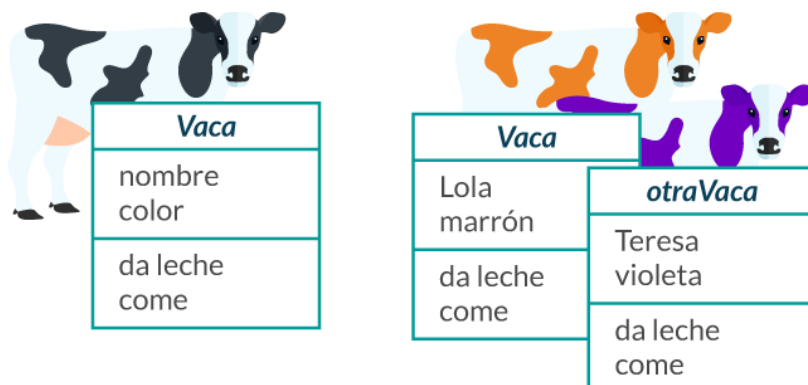
Introducción a la Programación
Orientada a Objetos (POO)

Clase
- atributo 1 : int - atributo 2 : String
+metodo1(parametro : int) : double +metodo2()

Algunas definiciones encontradas sobre que es un objeto y que nos pueden dar claridad del tema son:

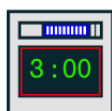
- “Un objeto se caracteriza por un número de operaciones y un estado que recuerda el efecto de estas operaciones”. [Ivan Jacobson](#)
- “Un objeto tiene un estado, comportamiento e identidad; la estructura y comportamiento de objetos similares se definen en sus clases comunes”. [Grady Booch](#)
- “Un objeto es una entidad que tiene un estado (cuya representación esta oculta) y un conjunto definido de operaciones que operan sobre ese estado”. [Ian Sommerville](#)
- “Un objeto es una identidad con unos límites bien definidos que encapsulan estado y comportamiento. El estado se representa por atributos y relaciones, el comportamiento es representado por operaciones y métodos”. [Object Management Group](#)

Los objetos pertenecen a una clase cumpliendo con la estructura de sus atributos. El atributo puede visualizarse en la definición de la clase y en las que heredan de la misma.





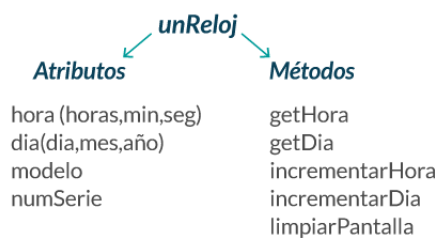
Modelo físico



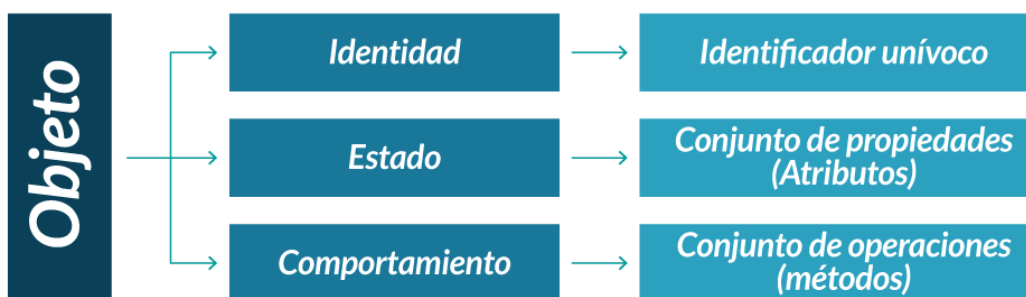
← Botón para mostrar la Hora

← Botón para mostrar la Día

Modelo informático



Los objetos poseen características como lo son la Identidad, el Comportamiento y el Estado. La Identidad es aquella que diferenciará a un objeto de otro y generalmente definirá el nombre al objeto. El comportamiento del objeto va ligado a la funcionalidad o responsabilidad, estableciendo las operaciones que el objeto ejecutará y también aquellas sobre las que puede dar una respuesta a mensajes recibido de otros objetos. La última característica llamada Estado corresponde al conjunto de atributos del objeto y sus valores en un momento dado.



La creación del objeto se hace mediante una instrucción new e ingresando el nombre de la clase de donde será una instanciada. En este momento se asignará un espacio de memoria para la creación del objeto donde almacenará los valores de sus atributos.



Lenguaje de programación



Lenguaje de Programación

Los lenguajes de programación son utilizados en la construcción de otros programas y están diseñados para organizar algoritmos y procesos lógicos que luego se ejecutarán en un dispositivo o sistema informático.

Estos lenguajes poseen símbolos y reglas sintácticas y semánticas, traducidas en instrucciones y relaciones lógicas, para construir el código fuente de un software determinado.

Posee características como lo son la simplicidad pues debe proveer conceptos simples para que el aprendizaje y su aplicación sea más sencilla no solo de comprender sino de mantener. Debe tener naturalidad en relación a que su aplicación cualquiera que sea, debe hacerse de forma natural, con estructuras, operadores y sintaxis que trabajen eficientemente. Características de abstracción para que posea la capacidad de definir y utilizar estructuras complicadas pasando por alto algunos detalles. Eficiencia en su traducción y ejecución y manejando de forma eficiente el espacio, el tiempo y la memoria. Debe ser estructurada y así disminuir la creación de errores. También deben tener Compacidad para expresar las operaciones de forma concisa y finalmente localidad.

Los lenguajes de programación pueden clasificarse de manera histórica por su estilo de programación en generaciones, por su nivel de abstracción, es decir dependiendo de la arquitectura de computación que se está utilizando, por paradigmas según su modelo y estilo de estructuración u organización de tareas, por propósito ya sea general o específico, por características comunes como su sintaxis, etc.

Tipos de datos



Tipos de Datos

En todo lenguaje de programación existe un conjunto de tipos de datos que representan las propiedades que tienen los datos definiendo su valor y las operaciones en los que estos se pueden emplear.

Estos pueden ser muchos y variados por lo que no es fácil decir cuántos tipos de datos hay pues incluso a través de estructuras de programación se pueden crear tipos de datos propios.



Existen unos tipos de datos que se conocen como primitivos y que son en general comunes entre todos los lenguajes de programación como lo son el char, int, long, float, double, boolean entre otros, y que se manejan en operaciones básicas como suma, resta, concatenación, etc. También hay tipo de datos abstractos que son generalidades de tipos de datos primitivos con procedimientos que generalizan las operaciones primitivas.



Métodos

Un método se puede entender como el comportamiento que tiene un objeto y como este actúa en su entorno o con otros objetos, en palabras simples define las funciones que el objeto puede realizar siendo la reunión de todos sus métodos los que definen el comportamiento general del objeto.

Para definir un método es necesario primero definir los atributos de las clases pues los métodos son

algoritmos de la clase o servicios que esta puede prestar a otras clases o a ella misma para una acción o solución particular.

Definimos los métodos como aquellas acciones que ejecutan los objetos especificando su comportamiento, se definen en una clase y pertenecen ya sea a una clase como a un objeto generalmente son una secuencia para ejecutar una acción, con parámetros de entrada que las regulan y en algunos casos un valor de retorno de algún tipo.





Los métodos se diferencian de los procedimientos y las funciones, porque está asociado directamente a un objeto o una clase y de este modo podría acceder y cambiar sus datos privados por tanto no se toma como una simple lista de instrucciones sino define la utilidad del objeto.

Se establecen algunos tipos de métodos como lo son métodos de instancia relacionados con un objeto en particular, métodos estáticos, de clase o compartidos asociados a una clase en particular, métodos de acceso que proveen información del estado de un objeto o cambiar incluso el estado de un objeto en cuyo caso se llamaría métodos de actualización o métodos de mutación.

El método se define a través de 4 elementos como lo son el nombre, parámetros como conjunto de valores con un tipo establecido que pueden resolver un problema puntual, un tipo de respuesta que determina el tipo de datos del resultado que devolverá el método y por último el cuerpo del método o lista de instrucciones que representa el algoritmo.

Instrucción de Retorno



Instrucción de Retorno

Como describimos en el punto anterior el método cuenta con unas instrucciones que representan el algoritmo y que resolverán un problema específico o prestará un servicio a otra u otras clases, y que el computador irá ejecutando una a una. Existe una instrucción sencilla que es la instrucción de retorno o return con la cual se informará el resultado que se dará como solución al problema.

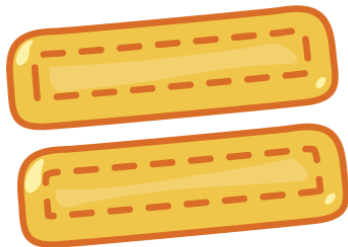
Inicialmente se debe declarar la clase definiendo cada uno de sus atributos y sus asociaciones, para pasar a la declaración de los métodos de la clase con su estructura correspondiente teniendo en cuenta que aquellos métodos que se usan en otras clases se deben definir como públicos. Esta definición incluye en el método las instrucciones necesarias para resolver el problema puntual planteado, pero en el cuerpo del método solo se hace referencia a los atributos del objeto. Los métodos que devolverán resultados deben especificar en su cuerpo de instrucciones la instrucción de retorno correspondiente.



Semana 1

Introducción a la Programación
Orientada a Objetos (POO)

Instrucción de Asignación



Instrucción de Asignación

Esta instrucción es para métodos que no tiene como fin calcular un valor, sino que modifican el estado del objeto. Se utiliza la instrucción de asignación (=) para definir el nuevo valor que tendrá un atributo en donde en el lado izquierda de la asignación se coloca el atributo a modificar y en la parte derecha el nuevo valor que debe guardará en el atributo.

Instrucción de Llamada de un Método



Instrucción de Llamada de un Método

Existen ocasiones en las que se necesita que un objeto invoque un método de otro objeto con el cual tiene una asociación, esto como parte de la solución. En esta llamada o al hacer referencia al objeto y al nombre del método, el computador busca al objeto y llama al método para darle la información a los parámetros.

Una vez se ejecuten las instrucciones traerá la respuesta si así corresponde. Esta instrucción en donde el objeto invoca un método de otro objeto con el

cual presenta una asociación se puede aplicar también entre métodos de la misma clase.

Llamada de Métodos con Parámetros



Llamada de Métodos con Parámetros

Un método requiere de parámetros en el momento en que la información que el objeto tiene es insuficiente para dar una solución, en este caso se define un tipo de parámetro asignándole un nombre que denote una idea clara del valor que se recibirá y este podrá ser usado solo usando en nombre del parámetro en el método.

Cuando se invoca este parámetro se pasarán de la misma manera sus valores, sean estos constantes, de



Semana I

Introducción a la Programación
Orientada a Objetos (POO)

atributos, expresiones, etc. El número de valores debe coincidir con el número de parámetros de lo contrario se genera un error.

Creación de Objetos



Creación de Objetos

Sobre este tema solo se hará referencia como un elemento del programa, pero el detalle de objetos y su creación se explicará más adelante a fondo, solo debemos entender que su creación inicia en la interfaz de usuario, cuando se crea una instancia de la clase y se almacena o referencia en cualquier atributo o variable que sea del mismo tipo.