



N° 1

Guía Didáctica: Algoritmo y Programación 2.

Competencia

Adquirir destrezas en el diseño y desarrollo de programas enfocados bajo el paradigma de la Programación Orientada a Objetos.

Contenidos

Concepto de POO, características, términos usados, fundamentos.

Conceptos relacionados con la POO.: Herencia, Encapsulamiento, Abstracción y Polimorfismo.

Evaluación

Trabajo de Investigación al final de la semana 3 (30%)

Taller práctico al final de la semana 4 (35%)

Taller práctico al final de la semana 6 (35%)

Docente: Víctor Hernández	Asignatura: Algoritmo y Programación 2
Lapso Académico:	1-2020
Carrera: Informática	Semestre: 3
Fecha de elaboración: 18 Mayo 2020	

Tema 1.

Introducción a la Programación Orientada a Objetos (POO).

Un paradigma se puede referir a una teoría o conjunto de teorías que sirve de modelo a seguir para resolver problemas o situaciones determinadas que se planteen. En el contexto de la programación, se establece como la forma en que un programador o un conjunto de programadores dan solución a uno o varios problemas claramente definidos. En este sentido, representa una manera particular de ofrecer soluciones.

La Programación Orientada a Objetos (POO) es un paradigma de programación (un modelo), la cual permite relacionar los objetos del mundo real y de alguna manera llevar estos conceptos a la programación.

Y como se puede relacionar los objetos del mundo real con la programación?, pues bueno lo que se intenta es crear sistemas basados no sólo en código sino más bien pensar en objetos que se comuniquen entre sí, reutilizar código y de alguna manera modularizar un programa.

Para entender mejor, imagina el cuerpo humano, tiene un conjunto de partes internas que necesita para funcionar, estas partes están agrupadas en sistemas, por ejemplo sistema nervioso, digestivo, linfático, respiratorio etc., cada uno tiene una estructura que agrupa funcionalidades específicas para formar un sistema, por ejemplo el sistema digestivo se encarga de la absorción y transformación de los alimentos. En fin, se plantea todo muy organizado, separado y agrupado de acuerdo cada funcionalidad.

Eso mismo es lo que trata de simular la POO en la programación, agrupar características similares llamadas clases (que pertenezcan a un mismo tipo), a partir de esas clases crear a objetos y que esos objetos se comuniquen entre sí mismo a través de mensajes, de tal manera que la unión de todos esos objetos forme un programa.

Eso mismo es lo que trata de simular la POO en la programación, agrupar características similares llamadas clases (que pertenezcan a un mismo tipo), a partir de esas clases crear a objetos y que esos objetos se comuniquen entre sí mismo a través de mensajes, de tal manera que la unión de todos esos objetos forme un programa.

La base de la programación orientada a objetos en Java son las clases y objetos, entre otros términos que también escucharás por ejemplo: instancia, constructores, variables de clase, variables instancia, métodos, herencia, encapsulamiento.

Tópicos a investigar:

1. Lenguaje JAVA: Introducción, origen y características.
2. Tecnologías relacionadas a JAVA: JDK, JVM, JRE, JNI, JAVA SE, JAVA EE.

Sitio de referencia que lleva este contenido relacionado: (Blog de la materia)

<https://alp2-1-2020.blogspot.com/>

Tema 2.

Elementos básicos en POO: Clase, Objeto, Instancia, Atributo, Método y Evento.

La base de la Programación Orientada a Objetos es el objeto. En la vida real todos los objetos tienen una serie de características y un comportamiento. Por ejemplo, una puerta tiene color, forma, dimensiones, material... (goza de una serie de características) y puede abrirse, cerrarse... (posee un comportamiento). En Programación Orientada a Objetos, un objeto es una combinación de unos datos específicos y de las rutinas que pueden operar con esos datos.

En relación a dichos datos, denominados Atributos, estos componentes del objeto también se les denomina variables miembro. Estos datos pueden ser de tipo primitivo (boolean, int, double, char...) o, a su vez, de otro tipo de objeto (lo que se denomina agregación o composición de objetos). La idea es que un atributo representa una propiedad determinada de un objeto.

Por otro lado, las rutinas, llamadas también Métodos, son los componentes del objeto que lleva a cabo una determinada acción o tarea con los atributos. En principio, todas las variables y rutinas de un programa de Java deben pertenecer a una clase. De hecho, en Java no hay noción de programa principal y las subrutinas no existen como unidades modulares independientes, sino que forman siempre parte de alguna clase.

Ahora, entrando en el concepto de Clase, se entiende como el conjunto de objetos que comparten una estructura y un comportamiento comunes. Una clase es una combinación específica de atributos y métodos y puede considerarse un tipo de dato de cualquier tipo no primitivo. Así, una clase es una especie de plantilla o prototipo de objetos: define los atributos que componen ese tipo de objetos y los métodos que pueden emplearse para trabajar con esos objetos.

Una instancia es un elemento tangible (ocupa memoria durante la ejecución del programa) generado a partir de una definición de clase. Todos los objetos empleados en un programa han de pertenecer a una clase determinada.

Aunque el término a veces se emplea de una forma imprecisa, un objeto es una instancia de una clase predefinida en Java o declarada por el usuario y referenciada por una variable que almacena su dirección de memoria. Cuando se dice que Java no tiene punteros simplemente se indica que Java no tiene punteros que el programador pueda ver, ya que todas las referencias a objeto son de hecho punteros en la representación interna.

En general, el acceso a los atributos se realiza a través del operador punto, que separa al identificador de la referencia del identificador del atributo (`idReferencia.idAtributo`). Las llamadas a los métodos para realizar las distintas acciones se llevan a cabo separando los identificadores de la referencia y del método correspondiente con el operador punto (`idReferencia.idMetodo(parametros)`).

Finalmente, un concepto valioso en la Programación es lo que se denomina un Evento, que se define como la comunicación o interacción entre dos objetos. En el caso de operaciones triviales como son las acciones del usuario sobre el programa, ahí se generan eventos. Son eventos típicos el click sobre un botón, el hacer doble click sobre el nombre de un fichero para abrirlo, el arrastrar un icono, el pulsar una tecla o combinación de teclas, el elegir una opción de un menú, el escribir en una caja de texto, o simplemente mover el ratón, por nombrar solo unos pocos.

Sitios de referencia relativos a POO: (sitios para consultar)

<https://ed.team/blog/que-es-la-programacion-orientada-objetos/>

<https://www.campusmvp.es/recursos/post/los-conceptos-fundamentales-sobre-programacion-orientada-objetos-explicados-de-manera-simple.aspx>

Sitio de referencia que lleva este contenido relacionado: (Blog de la materia)

<https://alp2-1-2020.blogspot.com/>

Tema 3.

Conceptos relacionados con POO: Abstracción, Encapsulamiento, Herencia y Polimorfismo.

En POO, la Abstracción consiste en ver a algo como un todo sin saber cómo está formado internamente. En este caso la idea es concebir el objeto como un todo, procurando enfocarse más en su desempeño externo que en su funcionamiento interno. Para entender mejor pensemos en un Teléfono inteligente, es un sistema muy complejo que a su vez está compuesto por otros subsistemas internos o partes que lo componen, por ejemplo: sistema operativo, aplicaciones, sensores, cámara, pero en definitiva el usuario lo ve como un todo, un Teléfono sin importar cómo está compuesto, qué versión tiene el kernel del sistema operativo, que controladores se debe instalar para utilizar un sensor, y así sucesivamente. Al usuario lo que le interesa es usarlo, saber que es un Teléfono inteligente y nada más, se abstrae de los detalles internos

En conclusión el concepto de abstracción trata la descomposición de sistemas complejos en partes sencillas, las cuales se pueden describir fácilmente.

Un ejemplo de abstracción en Java es una clase, ya que se la define como un todo. Por lo tanto, el objetivo es que el usuario pueda crear objetos de esa clase, que utilice sus métodos, y que no se preocupe por su implementación o como está diseñada.

Cuando se habla de Encapsulamiento, hablamos de ocultar la información, este concepto es un poco más claro que la abstracción. Bueno este tema significa que sólo se debe mostrar los detalles esenciales de un objeto, mientras que los detalles no esenciales se los debe ocultar.

Podemos hacer una analogía pensando en un sistema operativo Linux, en Linux si eres un usuario root tienes todos los permisos del sistema y puedes acceder a todas la configuraciones. Mientras que si eres sólo un usuario, solo tendrás acceso parcial al sistema (ocultación de información).

El encapsulamiento en Java lo tenemos en una clase, en la que se puede ocultar o restringir los datos (variables) o el código (métodos) utilizando palabras reservadas del lenguaje.

El concepto fundamental de la Herencia es el proceso en el que un objeto adquiere características de otro objeto. Siguiendo con el ejemplo anterior del celular, un celular es un equipo con características básicas. Ahora pensemos en un teléfono inteligente, que es un teléfono con

características muy avanzadas en cuanto al celular, pero que en sí guardan cierta relación. Yo podría crear un Smartphone desde cero, pero si te das cuenta, hay cosas que en el celular y el teléfono inteligente son comunes y que las podría reutilizar y no volver a crear de nuevo. El problema anterior se soluciona utilizando la herencia, con lo que ya no crearía uno nuevo si no que heredaría del objeto celular (reutilización) y lo que haría simplemente es añadir solo las nuevas características y tendría el nuevo modelo que es el teléfono inteligente. La herencia también ayuda a la reutilización, puesto que si ya tengo algo creado lo puedo reutilizar y no perder tiempo creando algo de nuevo.

En Java se maneja herencia simple, es decir que solo se puede heredar de una sola clase a la vez. Para decir que una clase hereda de otra, se utiliza la palabra reservada `extends` seguido de la clase que se quiere heredar.

El polimorfismo es una palabra de origen griego que significa “muchas formas”, es un término un poco complejo de entender, que a veces sólo se lo entiende con la práctica: el concepto dice que es la capacidad de un objeto para comportarse de diferentes formas de acuerdo al mensaje enviado.

Este término está relacionado con la herencia. Podemos hacer una analogía con el celular, un celular tiene la opción de llamada, tu puedes hacer una llamada y comunicarte con Pablo por ejemplo, haces otra llamada y te comunicas con Ana, si te das cuenta la función es la misma “hacer llamada” pero te comunicas con diferente persona. La función hacer llamada se comporta de forma diferente para cada número marcado, que este caso sería el mensaje que le envías a la función hacer llamada.

En Java un método con el mismo nombre puede comportarse de forma diferente de acuerdo al objeto recibido. El compilador en tiempo de ejecución detecta que forma tomar de acuerdo al objeto que recibe.

Sitios de referencia relativos a Herencia, en POO: (sitios para consultar)

<https://jarroba.com/herencia-en-la-programacion-orientada-a-objetos-ejemplo-en-java/>

<https://javadesdecero.es/poo/herencia-java-tipos-ejemplos/>

Sitio de referencia relacionado con Polimorfismo, en POO: (sitio para consultar)

<https://ifgeekthen.everis.com/es/polimorfismo-en-java-programaci%C3%B3n-orientada-objetos>

Sitio de referencia que expone acerca de la Abstracción en POO: (sitio para consultar)

<https://tinchicus.com/2019/05/23/java-abstraccion/>

Sitios referentes sobre el Encapsulamiento, y su relación con la abstracción: (sitios de consulta)

https://es.wikibooks.org/wiki/Fundamentos_de_programaci%C3%B3n/La_Abstracci%C3%B3n_y_el_Encapsulamiento

<https://www.youtube.com/watch?v=WhqmPWPdVQ>

Sitio de referencia que lleva este contenido relacionado: (Blog de la materia)

<https://alp2-1-2020.blogspot.com/>