

Taller de conceptos y principios de programación orientada a objetos GA4220501095-AA2-EV01.

PROGRAMACIÓN ORIENTADA A OBJETOS

Es un modelo de programación que permite que el software se organice alrededor de datos u objetos lo cual permite que el código se reutilizable ordenado y fácil de mantener.

Las principales características de este tipo de programación es que soporta objetos, que tienen un tipo o clase asociado, un objeto se puede definir como un campo de datos con atributos y comportamientos únicos. Su enfoque radica en la utilización de grandes programas y de alta complejidad, su uso facilita la detección de errores en el código, lo cual ayuda que se pueda modificar con mayor facilidad.

GLOSARIO

| TERMINO | CONCEPTO |
|------------------------|--|
| Código | Conjunto de datos o instrucciones que de datos de un programa. |
| Atributos | Define una propiedad de un objeto o archivo. |
| objetos | Entidad dentro de un lenguaje de programación. |
| Métodos | Bloque de código que alberga una serie de instrucciones. |
| Herencias | Mecanismo por el cual una clase permite heredar las características (atributos y métodos) de otra clase. |
| Clases | Actúa plantilla o plano para crear objetos. |
| Eventos | Acción que ocurre en la ejecución de un programa. |
| Encapsulamiento | El encapsulamiento es el proceso que permite agrupar datos y también operaciones que relacionan estos datos bajo una misma unidad lógica. |
| Abstracción | Medio por el cual se enfrentan los problemas complejos para extraer las características esenciales y posteriormente representar dichas características en objetos. |
| Polimorfismo | Característica que permite que una abstracción tome diferentes formas o diferentes comportamientos según el contexto. |

| | |
|----------------------------------|---|
| Modularidad | Propiedad que permite dividir una aplicación o sistema en partes más pequeñas, idealmente deben ser muy independientes (bajo acoplamiento) y altamente funcionales (alta cohesión). |
| Jerarquía | Clasificaciones y ordenaciones de las abstracciones de un problema, permiten comprender de forma general la estructura de un sistema y favorecen, en algunos casos, la reutilización de características y comportamientos de las abstracciones definidas. |
| UML | Herramienta para capturar mediante símbolos y diagramas comunicar la idea de un sistema. |
| Diagramas de clases | Tipo de diagrama estático que muestra las clases los atributos y como se relacionan entre ellos. |
| Herramientas Case | Herramienta utilizada para el desarrollo y mantenimiento de sistemas informáticos. |
| Control de Acceso | Define la accesibilidad para los atributos y métodos los cuales pueden ser públicos, privado o protegidos. |
| Control de Acceso | Define la accesibilidad para los atributos y métodos los cuales pueden ser públicos, privado o protegidos. |
| Asociaciones | Relación que se establece entre dos o más clases o entidades, indicando cómo interactúan o se conectan entre sí. |
| Agregación | Tipo especial de asociación donde una clase hace parte de la otra. |
| Composición | Forma fuerte de agregación en la que una clase posee otra y hace que sus existencias dependen entre sí. |
| Patrones de Diseño | Soluciones reutilizables que en si no son códigos de programación si no guías para corregir problemas de arquitectura. |
| Case | Es una sigla, que corresponde a las iniciales de: Computer Aided Software Engineering; y en su traducción al español significa Ingeniería de Software Asistida por Computación. |
| Patrones comportamentales | Se centra en la interacción y comunicación entre los objetos de un sistema. |

| | |
|---------------------------------|--|
| Estrategia | Tipo de patrón que sirve para encapsular algoritmos lo que permite poder seleccionarlos de manera dinámica durante su ejecución. |
| Comando | Tipo de patrón que encapsula una solicitud como un objeto, lo que permite parametrizar clientes con diferentes solicitudes, colas de solicitudes o realizar un seguimiento de las operaciones. |
| Invoker | Elemento usado por los clientes y que le solicita al comando llevar a cabo una acción. |
| Cliente | Invoca la ejecución de las acciones desde el Invoker. |
| IComando | Interfaz donde se especifican las operaciones a ejecutar. |
| Receptor | Clase que realiza la acción. |
| Ejecutar | Operación que necesita ser llevada a cabo. |
| Iterador | Tipo de patrón orientado al trabajo con colecciones y facilita el acceso a todos los elementos de la colección sin tener la necesidad de conocer su estructura. |
| Patrones creacionales | Se enfocan en la manera que se crean los objetos en el software. |
| Patrones estructurales | Facilitan el diseño de relaciones entre entidades, promoviendo la eficiencia, la flexibilidad y la reutilización del código. |
| Fachada | Tipo de patrón que se utiliza cuando el sistema está compuesto por varios subsistemas y se hace complejo gestionar los mensajes que debe realizar el cliente en cada uno de estos subsistemas. |
| Delegate | Tipo de patrón que busca evitar asumir todas las responsabilidades en una sola instancia y delegar las actividades en otras instancias que son especializadas en resolver dicha tarea. |
| Vistas estáticas | No modela el comportamiento del sistema ni muestra las variaciones que se puedan presentar por efecto del tiempo. |
| Diagrama de objetos | Como su nombre lo dice son diagramas orientados a objetos. |
| Diagramas de componentes | Tipo de diagrama perteneciente a UML ayudan a identificar que componentes son necesarios para un producto en particular. |

| | |
|---------------------------------------|--|
| Nodos | Representa un elemento que puede ser hardware o software y se representa por un cubo. |
| Artefactos | Representan elementos concretos generados en el proceso de desarrollo, como por ejemplo bibliotecas, archivos, etc. |
| Asociación de comunicación | Representa el camino de comunicación entre los nodos y se representa por una línea continúa uniendo los nodos en cuestión. |
| Dispositivos | Es un tipo especial de nodo que representa un recurso computacional del sistema como por ejemplo un servidor. |
| Especificaciones de despliegue | Representan configuraciones que se deben tener en cuenta para desplegar un artefacto en un nodo. |