

Análisis y diseño orientado a objetos. Concepto. Elementos. El proceso unificado de software. El lenguaje de modelado unificado (UML)

Análisis y Diseño Orientado a Objetos (ADOO):

Concepto: El Análisis y Diseño Orientado a Objetos (ADOO) es un enfoque de desarrollo de software que se basa en la representación de sistemas como conjuntos de objetos interconectados, en lugar de tratarlos como procesos lineales y funciones. Se centra en modelar el mundo real a través de objetos, que son representaciones abstractas de entidades y conceptos del dominio de la aplicación. Este enfoque busca crear sistemas más comprensibles, reutilizables y mantenibles.

Elementos: El ADOO implica varios elementos clave:

1. **Objetos:** Los objetos son las unidades fundamentales del ADOO. Representan entidades del mundo real o conceptos abstractos y tienen atributos y comportamientos asociados. Por ejemplo, en un sistema de gestión de bibliotecas, un objeto «Libro» tendría atributos como título y autor, y métodos como «prestar» y «devolver».
2. **Clases:** Las clases son plantillas o modelos para crear objetos. Definen la estructura y el comportamiento común de un conjunto de objetos relacionados. Por ejemplo, una clase «Libro» definiría los atributos y métodos que todos los libros comparten.

3. **Herencia:** La herencia permite que las clases hereden atributos y comportamientos de otras clases. Esto fomenta la reutilización de código y la creación de jerarquías de clases. Por ejemplo, una clase «Libro de No Ficción» podría heredar de la clase «Libro».
4. **Encapsulación:** La encapsulación es el concepto de ocultar los detalles internos de una clase y exponer solo la interfaz necesaria para interactuar con los objetos. Esto promueve la modularidad y facilita el mantenimiento.
5. **Polimorfismo:** El polimorfismo permite que objetos de diferentes clases respondan de manera uniforme a una interfaz común. Esto permite tratar a diferentes objetos de manera similar y facilita la extensión del sistema.

El Proceso Unificado de Software:

El Proceso Unificado de Software (Rational Unified Process, RUP) es un enfoque de desarrollo de software que utiliza principios del AD00. RUP es un proceso iterativo y dirigido por casos de uso que se centra en la arquitectura, los requisitos y el diseño del sistema. Se divide en fases, cada una de las cuales produce entregables específicos. Las fases típicas de RUP incluyen:

1. **Inception (Inicio):** Definición del alcance, los objetivos y los riesgos del proyecto.
2. **Elaboration (Elaboración):** Definición de la arquitectura del sistema y los casos de uso clave.
3. **Construction (Construcción):** Implementación de componentes y funcionalidades del sistema.
4. **Transition (Transición):** Pruebas y despliegue del sistema.

RUP promueve el uso de modelos visuales para representar el

sistema, lo que nos lleva al siguiente punto: el Lenguaje de Modelado Unificado (UML).

Lenguaje de Modelado Unificado (UML) :

El UML es un lenguaje de modelado visual que se utiliza para representar sistemas orientados a objetos. Proporciona una notación estándar para describir estructuras, comportamientos, interacciones y relaciones en un sistema. Los elementos clave de UML incluyen:

1. **Diagramas UML:** UML incluye varios tipos de diagramas, como diagramas de clases, diagramas de secuencia, diagramas de casos de uso y muchos más. Cada tipo de diagrama se utiliza para representar aspectos específicos del sistema.
2. **Clases:** En UML, las clases se representan en diagramas de clases y se utilizan para definir la estructura y el comportamiento de objetos relacionados.
3. **Casos de Uso:** Los diagramas de casos de uso describen las interacciones entre actores externos y el sistema, mostrando cómo se utiliza el sistema para lograr objetivos específicos.
4. **Relaciones:** UML define relaciones como asociación, herencia, agregación y composición para modelar las conexiones entre objetos y clases.
5. **Diagramas de Secuencia:** Estos diagramas muestran cómo los objetos interactúan a lo largo del tiempo, lo que es útil para representar flujos de trabajo y procesos.

El UML es ampliamente utilizado en el desarrollo de software para ayudar a los equipos a comprender, comunicar y documentar sistemas complejos. Facilita la visualización y la documentación de los aspectos clave de un sistema basado en objetos y es una herramienta esencial en el enfoque del AD00 y

procesos como el RUP.