

Temas de Examen Unidad 1:

¿Qué son los Paradigmas?

Se puede definir al término paradigma como la forma de visualizar e interpretar los múltiples conceptos, esquemas o modelos del comportamiento en diversas disciplinas.

Paradigmas de Programación

Un paradigma de programación provee y determina la visión y métodos de un programador en la construcción de un programa o subprograma.

Clasificación de Paradigmas de Programación:

Paradigma Imperativo

Secuencia de instrucciones o comandos que cambian el estado de un programa

Paradigma Declarativo

Se basa en como describe (declara) cómo es algo. Es más complicado de implementar que el paradigma imperativo, tiene desventajas en la eficiencia, pero ventajas en la solución de determinados problemas.

Paradigma Estructurado

Se divide en bloques (procedimientos y funciones) que pueden o no comunicarse entre sí. Además, la programación se controla con secuencia, selección e iteración.

Paradigma Funcional

Hace hincapié en la aplicación de las funciones y composición entre ellas, más que en los cambios de estados y la ejecución secuencial de comandos (como lo hace el paradigma procedimental).

Paradigma lógico

Se basa en los conceptos de lógica matemática. Es un tipo de paradigma de programación declarativo. Define reglas lógicas para luego, a través de un motor de inferencias lógicas, responder preguntas planteadas al sistema y así resolver los problemas.

Paradigma Orientado a Objetos

Intenta simular el mundo real a través del significado de objetos que contienen características y funciones. Basado en la idea de encapsular estado y operaciones en objetos. Su principal ventaja es la reutilización de código y su facilidad para pensar soluciones a determinados problemas.

¿Qué es una Clase?

Una clase es un conjunto de objetos que comparten una estructura común y un comportamiento común.

Diferencia entre Clase y Objeto

Mientras que un objeto es una entidad concreta que existe en el espacio y en el tiempo, una clase representa sólo una abstracción, la "esencia" de un objeto.

Abstracción

Es la acción de separar propiedades. Se separan características esenciales de las que no lo son tanto. Se separa el "qué" debe hacerse del "cómo" se hará.

Interfaz e implementación

La interfaz de una clase proporciona su visión externa y enfatiza la abstracción a la vez que oculta su estructura y los secretos de su comportamiento.

Se puede dividir la interfaz de una clase en tres partes:

- Pública: Una declaración accesible a todos los clientes
- Protegida: Una declar. acces. sólo a la propia clase, sus subclases y sus clases amigas.
- Privada: Una declaración accesible sólo a la propia clase.

¿Qué es un Objeto?

Desde el punto de vista de los humanos un objeto es cualquiera de estas cosas:

- Una cosa tangible y/o visible.
- Algo que puede comprenderse intelectualmente.
- Algo hacia lo que se dirige un pensamiento o una acción.

"Representa un elemento, unidad o entidad individual e identificable, ya sea real o abstracta, con un rol bien definido en el dominio del problema".

Un objeto es una instancia de una clase. Los objetos pertenecen siempre a una clase, de la que toman su estructura y comportamiento.

"Un objeto tiene un estado, comportamiento e identidad; la estructura y comportamiento de objetos similares están definidos en su clase común; los términos instancia y objeto son intercambiables".

Estado

El estado de un objeto abarca todas las propiedades (normalmente estáticas) del mismo más los valores actuales (normalmente dinámicos) de cada una de esas propiedades.

- Objetos: existen en el tiempo, son modificables, tienen estado, son instanciados y pueden crearse, destruirse y compartirse.
- Valores simples: son atemporales, inmutables y no instanciados.

Comportamiento

"El comportamiento es cómo actúa y reacciona un objeto, en términos de sus cambios de estado y paso de mensajes".

El comportamiento de un objeto representa su actividad visible y comprobable exteriormente.

"El estado de un objeto representa los resultados acumulados de su comportamiento".

Identidad

"La identidad es aquella propiedad de un objeto que lo distingue de todos los demás objetos".

Relaciones entre objetos

La relación entre dos objetos cualesquiera abarca las suposiciones que cada uno realiza acerca del otro, incluyendo qué operaciones pueden realizarse y qué comportamiento se obtiene. Hay dos tipos de jerarquías de especial interés: Enlaces y Agregación.

Enlaces

Asociación específica por la cual un objeto (cliente) utiliza los servicios de otro objeto (servidor), o a través de la cual un objeto puede comunicarse con otro.

Visibilidad

La visibilidad es una propiedad que permite a un objeto operar sobre otro. Si un objeto no ve a otro, no puede enviarle un mensaje, para pedirle su colaboración.

Agregación

La agregación denota una jerarquía todo/parte, con la capacidad de ir desde el todo (agregado) hasta las partes. La agregación es a veces mejor porque encapsula partes y secretos del todo. A veces, son mejores los enlaces porque permiten acoplamientos más débiles entre los objetos.

Relaciones entre clases:

Asociación (conexión entre clases)

Una asociación denota una dependencia semántica y puede establecer o no, la dirección de esta dependencia, conocida como navegabilidad.

Generalización/Especialización (relaciones de Herencia)

"La herencia es una herramienta que permite definir nuevas clases en base a otras clases".

Polimorfismo

Concepto aplicado en el contexto de la teoría de tipos, como en las relaciones de herencia o generalización, donde el nombre de un método puede implicar comportamientos diferentes, conforme esté especificado en clases diferentes, en tanto y en cuanto estas clases estén relacionadas por una superclase común. Es la propiedad que potencia el uso de la herencia, las clases deben especificar los comportamientos con los mismos protocolos.

Agregación

Tienen un paralelismo directo con las relaciones de agregación entre los objetos correspondientes a esas clases. La agregación establece una dirección en la relación todo/parte. Tipos de contención:

- Contención física también conocida como composición: es una contención por valor, lo que significa que el objeto no existe independientemente de la instancia que lo encierra. El tiempo de vida de ambos objetos está en íntima conexión.
- Contención por referencia: es un tipo menos directo de agregación, en este caso se pueden crear y destruir instancias de cada objeto en forma independiente.

El Modelo de Objetos

Abstracción

"Una abstracción denota las características esenciales de un objeto, que lo distingue de todos los demás y proporciona así fronteras conceptuales nítidamente definidas respecto a la perspectiva del observador".

Encapsulamiento

"El encapsulamiento es el proceso de almacenar en un mismo compartimento los elementos de una abstracción, que constituyen su estructura y su comportamiento; sirve para separar la interfaz contractual de una abstracción y su implantación".

Modularidad

"La modularidad es la propiedad que tiene un sistema que ha sido descompuesto en un conjunto de módulos cohesivos y débilmente acoplados".

Jerarquía

"La jerarquía es una clasificación u ordenación de abstracciones".

Tipos (Tipificación)

"Los tipos son la puesta en vigor de la clase de objetos, de modo que los objetos de tipos distintos no pueden intercambiarse o, como mucho, pueden intercambiarse sólo de formas muy restringidas".

Concurrencia

"La concurrencia es la propiedad que distingue un objeto activo de uno que no está activo".

Persistencia

"La persistencia es la propiedad de un objeto por la que su existencia trasciende el tiempo".

Clasificación

La clasificación es el medio por el que ordenamos, el conocimiento ubicado en las abstracciones.

- La clasificación ayuda a identificar jerarquías de generalización-especialización y agregación entre clases.
- La clasificación también proporciona una guía para tomar decisiones sobre modularización. Se puede decidir ubicar ciertas clases y objetos juntos o en módulos diferentes, dependiendo de su similitud. El acoplamiento y la cohesión son simplemente medidas de esta similitud.
- La clasificación desempeña un papel en la asignación de procesos a los procesadores. Se ubican ciertos procesos juntos en el mismo procesador o en diferentes procesadores, dependiendo de intereses de empaquetamiento, eficacia o fiabilidad.

El modelado en el Desarrollo de Software

¿Qué es un modelo?

Un modelo es una simplificación de la realidad.

- Los modelos nos ayudan a visualizar cómo es o queremos que sea un sistema.
- Los modelos nos permiten especificar la estructura o el comportamiento de un sistema.
- Los modelos proporcionan plantillas que nos guían en la construcción de un sistema.
- Los modelos documentan las decisiones que hemos adoptado.

Lenguaje de Modelado Unificado (UML)

Es un lenguaje de modelado, de propósito general, usado para la visualización, especificación, construcción y documentación de sistemas Orientados a Objetos

- Lenguaje: el UML es, precisamente, un lenguaje, lo que implica que éste cuenta con una sintaxis y una semántica. Al modelar un concepto en UML, existen reglas sobre cómo deben agruparse los elementos del lenguaje y el significado de esta agrupación.
- Modelado: el UML es visual. Mediante su sintaxis se modelan distintos aspectos del mundo real, que permiten una mejor interpretación y entendimiento de éste.
- Unificado: unifica varias técnicas de modelado en una única.

Diagrama de Clases

Los diagramas de clases muestran una vista estática de la estructura del sistema, o de una parte de éste, describiendo qué atributos y comportamiento debe desarrollar el sistema, detallando los métodos necesarios para llevar a cabo las operaciones del sistema. El diagrama de clases muestra un conjunto de interfaces, colaboraciones y sus relaciones.

Componentes del Diagrama de Clases

Visibilidad: normalmente se desea ocultar los detalles de implementación y mostrar sólo las características necesarias para llevar a cabo las responsabilidades de la abstracción. Hay tres niveles disponibles:

- Pública: Cualquier clasificador externo con visibilidad hacia el clasificador dado puede utilizar la característica. Se representa con el símbolo "+".
- Protegida: Cualquier descendiente del clasificador puede utilizar la característica. Se representa con el símbolo "#".
- Privada: Sólo el propio clasificador puede utilizar la característica. Se representa con el símbolo "-".
- Paquete: Visible para clasificadores del mismo paquete. Se representa con el símbolo "~".

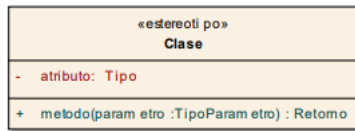


Fig.21- Notación para una clase en UML

Clases de Análisis

UML, ha incorporado como parte del lenguaje el uso de tres tipos de clases, denominadas clases de análisis, cuyo propósito es lograr una estructura más estable y mantenible del sistema que será robusta para el ciclo de vida entero del sistema.

Clase de Entidad

La clase de entidad modela información en el sistema que debería guardarse por mucho tiempo y, comúnmente, debería sobrevivir la ejecución del caso de uso del sistema en la que se creó.

Clase de Interfaz

La clase de interface modela el comportamiento e información que es dependiente de la frontera del sistema con el ambiente.

Clase de Control

La clase de control modela funcionalidad que implica operar sobre varios objetos diferentes de entidad, haciendo algunos cálculos y retornando el resultado al objeto de interface

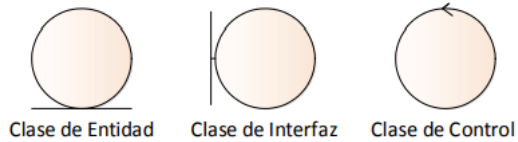


Fig. 22- Representación icónica de los tres tipos de clases de UML

Interface

Una interface es un tipo especial de clase que agrupa una colección de operaciones que especifican un servicio de una clase o componente



Fig. 23- Notaciones para una Interface en UML, representación etiquetada e icónica

Existen dos tipos de interfaces:

- Interfaces Requeridas: Muestran lo que el clasificar al que pertenecen puede requerir del ambiente a través de un puerto dado.
- Interfaces Provistas: Muestra la funcionalidad que expone un clasificador a su ambiente.

Relaciones

Una relación es una conexión entre dos elementos. En el contexto del diagrama de clases, los elementos que se relacionan son clases e interfaces. Los tipos de relaciones son:

Asociación: Relación estructural que especifica que los objetos de un elemento se conectan a los objetos de otro elemento. Este tipo de relación se suele implementar con una referencia en uno de los dos elementos que participan de la relación hacia el otro elemento. Estas relaciones pueden incluir multiplicidad, navegabilidad y el nombre del rol que se establece en la relación. Al implementarse esta relación el rol de la asociación suele ser del tipo de uno de los elementos en la relación.

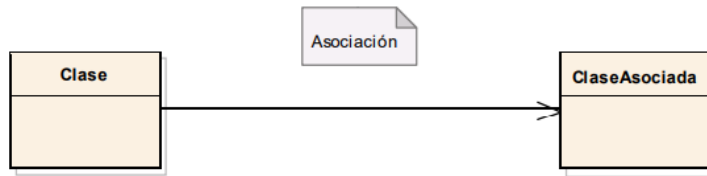


Fig. 25- Notación para representar la relación de asociación entre clases en UML

Agregación: Es un tipo especial de asociación, que representa una relación completamente conceptual entre un "Todo" y sus "Partes".

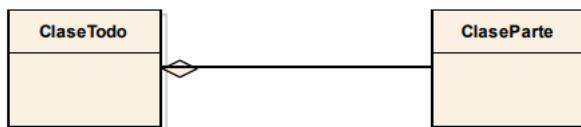


Fig. 26- Notación para representar la relación de agregación entre clases en UML

Composición: Es una variación de la agregación que muestra una relación más fuerte entre el todo y sus partes. Se utiliza cuando existe una relación de contenedor-contenido entre dos 66 66 elementos. Usualmente una instancia de una parte suele pertenecer sólo a una instancia del todo. Esta relación en general implica que al borrarse el todo se borran todas sus partes.

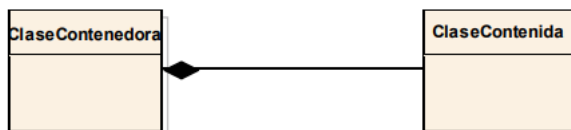


Fig. 27- Notación para representar la relación de composición entre clases en UML