**ENTORNO DE DESARROLLO**

**Curso 2018-19**

**PRÁCTICA 4. DIAGRAMAS DE CLASES**

**Introducción al marco teórico**

Responde a las siguientes cuestiones: ¿Qué es UML? ¿Qué son los diagramas de estructura? ¿Y los de comportamiento? ¿Y los de iteracción?

Busca un ejemplo en internet de diagrama de clases, diagrama de casos de uso y diagrama de secuencia. Adjúntalo.

Que es una clase y que tipos de diagramas existen

¿Que es **UML**?

El **Lenguaje Unificado de Modelado** (**UML**) es un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software, p. ej., en el flujo de procesos en la fabricación.

UML no es un lenguaje de programación es un lenguaje de modelado, es para crear diseños lógicos, UML es el lenguaje actual mas usado para el modelado de software, se construye de manera lógica no física

¿Que son los diagrama de casos de uso?

Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema.

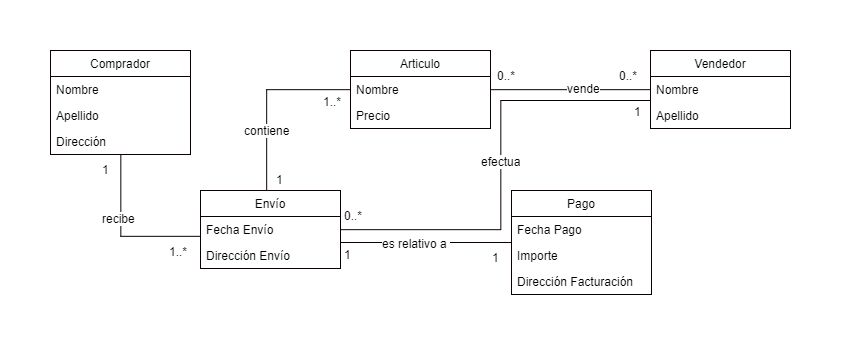
¿Que son los diagramas de comportamiento?

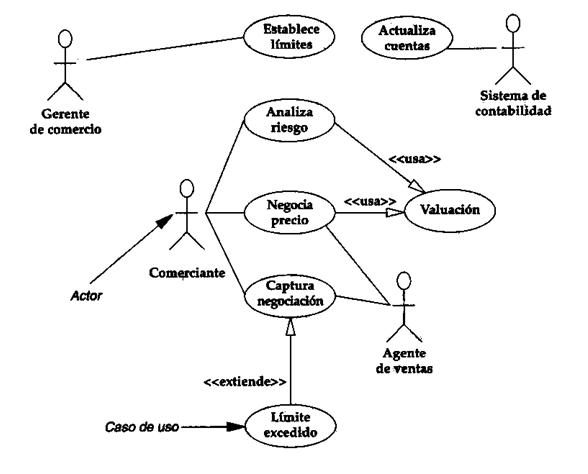
Especificar o de detallar la estructura de los elementos del sistema, se llaman **diagramas estructurales**. No indican el movimiento de dato y se les llama diagramas de representación estática.

Como se va comportando el sistema, indican el movimiento de los datos por el sistema llamados **diagramas de comportamiento** y se les llama también diagramas dinámicos

**Diagramas de iteraccion:** como interactúan los diagramas entre ellos.

**Diagrama de clase**

**Diagramas de caso de uso**

Laasociaciónentre dos clases: cadinalidades (muchas) y se escribe 1:\*, con un 1 en cuanto se instancie es obligatorio decir que se esta instanciando y 0 que se instancia en el tiempo

Cadinalidad minima 1

navegabilidad: sentido unidireccional(1:\*) o bidireccional

**Diagramas de clase**

Es un tipo de diagrama estático de estructuras que describe la estructura del sistema mostrando sus clases y las asociaciones entre ellas.

**Clases. Definición y representación. Los atributos y su visibilidad. Los métodos y su visibilidad**

Una clase es una unidad básica que encapsula toda la información de un objeto (instancia de una clase). Un atributo representa alguna propiedad de la clase que se encuentra en todas las instancias de la clase. Se puede representar con su tipo e incluso con su valor por defecto. Pueden ser públicos (+), privados (-), protegidos (#) o empaquetados ().

Representar la clase de producto: Tiene un código, una descripción, un precio de venta, una cantidad en el almacén y una cantidad mínima de pedido. Se hace necesario calcular el stock, el precio venta público, conocer el código y la descripción del producto.

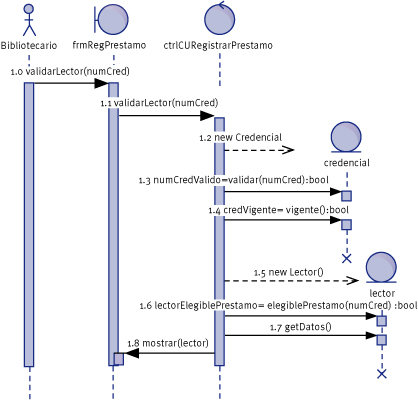
**Relaciones, sus asociaciones y su cardinalidad. Navegabilidad**

Representar las relaciones siguientes:

Una asignatura solo es impartida por un profesor. Los alumnos se matriculan de diversas asignaturas. Una asignatura pertenece a un ciclo formativo o a varios.

**Clase reflexiva**

**Diagrama de secuencia**

**Diagrama de clase representa como estan estructurados los datos y como interactuan entre ellos, asociacion**

Cuando una clase puede asociarse consigo misma.

Pon un ejemplo de una clase reflexiva y represéntalo.

**Clase asociación**

Cuando una asociación entre dos clases lleva información necesario para esa asociación.

Mostrar la clase asociación compra, donde un cliente compra muchos productos, y este puede ser comprado por muchos clientes, y de la compra se necesita saber la fecha de compra y las unidades compradas.

Se le llama asociacion a la relacion entre dos clases

**Generalización y especialización**

Las clases con atributos y/o métodos comunes se pueden organizar de forma jerárquica mediante la herencia (las subclases heredan de la superclase). El código queda public class alumno extends persona{…..}

Crea la clase datos personales, alumnos y profesores representando la jerarquía de sus atributos y operaciones.

Indicar el código java generado para estas clases

**Composición**

Un objeto puede estar compuesto por otros objetos que lo constituyen y no pueden ser compartidos por varios objetos compuestos (cardinalidad máxima siempre es 1). La supresión del objeto compuesto comporta la supresión de los componentes.

Crea la clase ordenador con sus atributos e indica las diferentes clases que intervienen en la composición: placabase, memoria, teclado, almacenamiento. Indica cardinalidad mínima y máxima.

Indica el código java generado para la clase ordenador, indicando los métodos getter y setter.

**Agregación**

Los componentes pueden ser compartidos por varios compuestos y la destrucción no implica la destrucción de los componentes.

Pon un ejemplo de agregación, indicando cardinalidad mínima y máxima.

METODOLOGÍA

Para poder realizar diagramas UML usando Eclipse instalaremos el plugin UML2.

Para ello, help-Eclipse Marke Place- tecleamos UML2 en find. Elegimos el plugin que corresponde con la versión eclipse (UML Designer).

Se ha creado la vista Modeling. Para crear un proyecto abrimos File-New-UML Project.

Nos posicionamos sobre el proyecto y en el menú contextual se elige Create Representation. En diagramas de estructuras o estructurales (UML Structural Modelin) se encuentra el diagrama de clases (Class Diagram).

Se selecciona en qué modelo se creará el diagrama, se teclea un nombre y se abre la vista de diseño.

Captura la pantalla e indica las distintas zonas de esta vis: explorador del modelo, vista de diseño, pestaña de propiedades para cambiar las características de los elementos y la paleta de elementos que se pueden añadir al diagrama.

Al crear una clase si se hace doble click podemos elegir la visibilidad y el tipo de clase: Is Abstract (clase abstracta), is leaf (hoja, es decir, no podrá ser una especialización), Is Active (si es activa). También se puede asignar un caso de uso.

El las pestañas Attributes y Operations, podremos gestionar las propiedades y métodos de la clase. Las propiedades de un atributos puede ser: is read only (no puede cambiar su valor), is istatic (las instancias de este tipo comparten el mismo valor para ese atributo), is leaf (no se permite que este atributo se redefina en los tipos derivados), is derived (se calcula a partir de otros atributos), is ordered (colección secuencia y ordenada), is unique (no hay valores duplicados para la colección).

Para los tipos de datos se seleccionará de la lista PrimitiveType. Si no hay tipos de datos se hace necesario importarlos: en el menú contextual sobre el proyecto se elige Import primitive types (elegir Java Primitive types).

Para los métodos se teclea el nombre, la visibilidad y se marca el tipo, añadiendo los parámetros si los tiene (en la pestaña parameteres). Se selecciona el tipo de dato y se marca el tipo de parámetro, in de entrada, inout de entrada/salida y out de salida. Se indica return si la operación devuelve un valor de retorno.

También se puede cambiar las propiedades de la clase desde Windows/Show view/Properties.

Cambiaremos las propiedades de semántica en la asociación para indicar la cardinalidad mínima y máxima (en property). El la propiedad lower se indica la cardinalidad mínima y en upper la máxima.

**PRÁCTICA 4.1**

Se trata de realizar un diagrama de clases para representar las relaciones entre empresa, empleados y clientes. Usaremos asociaciones de composición y de generalización en el diagrama. Los requisitos son:

La empresa se compone de clientes y de empleados (usamos composición).

Datos de la empresa son CIF, razón social, dirección y teléfono.

Datos de clientes son código de cliente, nombre, fecha de nacimiento, teléfono, empresa para la que trabaja y comisión.

Datos de empleados son código de empleado, nombre, fecha de nacimiento, teléfono, fecha de alta en la empresa y salario.

Un empleado puede ser director de varios empleados. Se necesita saber la categoría y la fecha de alta como director.

Para todas las clases crearemos dos métodos, uno para asignar datos a los atributos (como si fuese el constructor) teniendo tantos parámetros como atributos, y otro que se llama obtener que devolverá un objeto de la misma clase.

**PRÁCTICA 4.2**

Cambiar el diseño anterior atendiendo que se desea almacenar los teléfonos, los correos electrónicos y el domicilio. Modela el domicilio.

**PRÁCTICA 4.3**

Se desea realizar el análisis de un sistema de gestión informática de un centro de formación. El sistema debe proporcionar una ventana inicial con una serie de menús que abrirán paso al resto de ventas de la aplicación. Las ventas deben permitir realizar las siguientes acciones:

La gestión de la matrícula de alumnos en los cursos, para dar de alta, eliminar, modificar o consultar datos de matriculación.

La gestión de cursos, para mantener actualizados sus datos se realizarán operaciones de consulta, alta, baja y modificación, y además generar listados de cursos y alumnos de cursos.

Disponemos de una base de datos donde están almacenados los datos de los cursos, los alumnos y las matrículas. Los datos son los siguientes:

Los datos de curso son código, descripción, horas y precio (el precio puede variar a lo largo de la vida de la aplicación).

Los datos de alumnos son código, nombre, fecha de nacimiento y dominicilio

Los datos matricula son el número de matrícula y la fecha.

Evidentemente un alumno se puede matricular en sendos cursos y en estos, obviamente, se matricularán varios alumnos.

Los cursos son de dos tipos: preparatorios y propedéuticos. Se hace necesario conocer de los segundos cuáles son sus primeros.

REALIZAR:

Identificar las clases de diseño y sus relaciones: clases de tipo entidad, de control e interfaz

Identificar los atributos y las operaciones.

Identificar los paquetes: una para las clases entidad, otro para las clases de control y otros para las clases interfaz.

Realizar el diagrama de clases y de paquetes.

Genera el código.