PREGRADO

UNIDAD 1: OVERVIEW

UML CLASS DIAGRAMS

SI720 | Diseño y Patrones de Software

Al finalizar la unidad, el estudiante elabora y comunica artefactos de diseño de software aplicando principios básicos y patrones de diseño para un dominio y contexto determinados

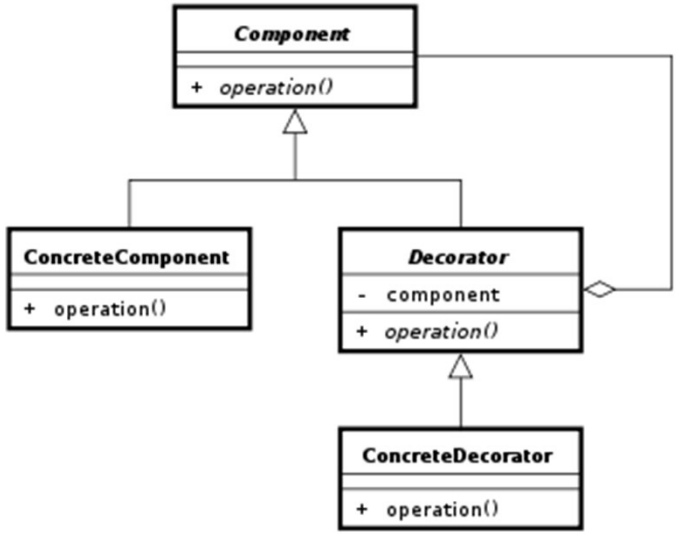
AGENDA

CLASS DIAGRAMS

REFLEXIONES

Class Diagram

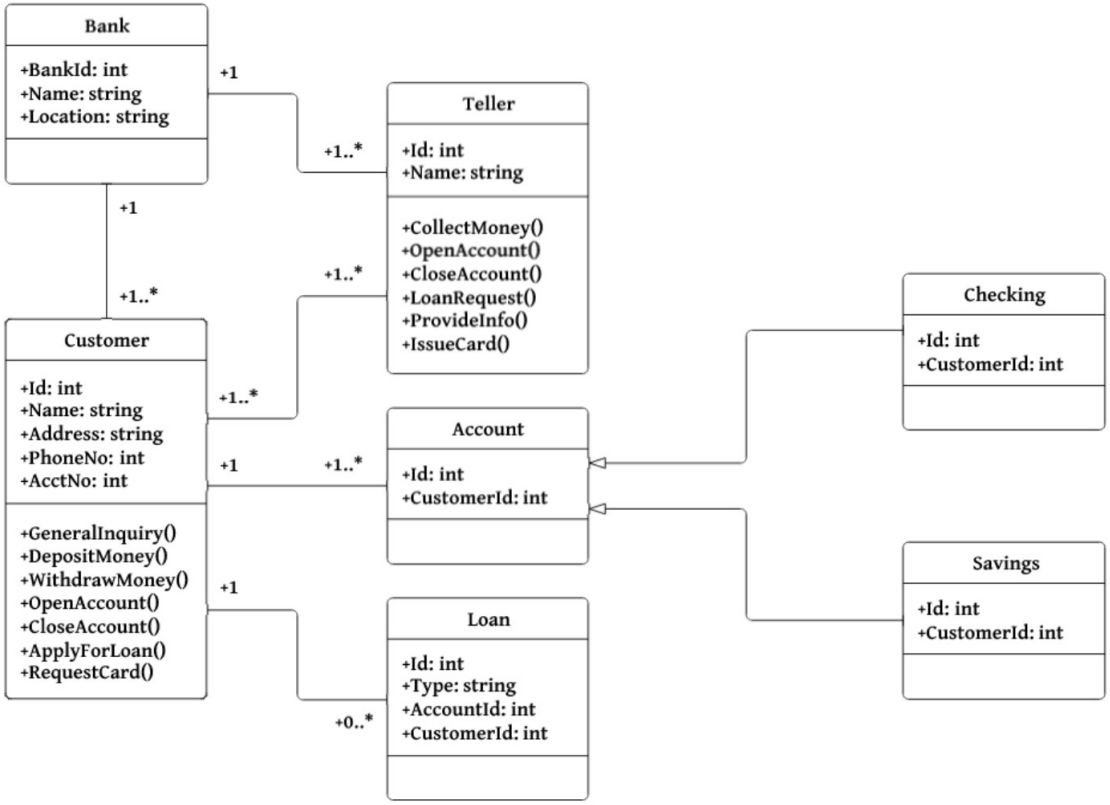
Los patrones que encontramos deben ser capturados y documentados de manera suficientemente descriptiva para que puedan ser referidos para uso futuro.
UML proporciona las herramientas perfectas para hacer precisamente esto.
El diagrama de clases en UML se puede utilizar para capturar los patrones identificados en un sistema. Además, UML tiene un vocabulario suficientemente extenso y expresivo para capturar los detalles de los patrones.



Podremos utilizar el diagrama de clases para describir el problema y la solución con el patrón.

What is a Class Diagram?

Los diagramas de clase son una forma ordenada de visualizar las clases en su Sistema, antes de comenzar a codificarlas. Son una representación estática de la estructura de su sistema.



* Example of a Class Diagram for a Banking System

Why Do We need Class Diagrams?

- Planificar y modelar con anticipación hace que la programación sea mucho más fácil.
- Hacer cambios en los diagramas de clase es fácil, mientras que codificar diferentes funciones después del hecho no es práctico.
- Permite tener un plan, un plan de diseño, para poder ANALIZAR y modificar su sistema.
- No necesita muchos conocimientos técnicos / específicos del idioma para comprenderlo.

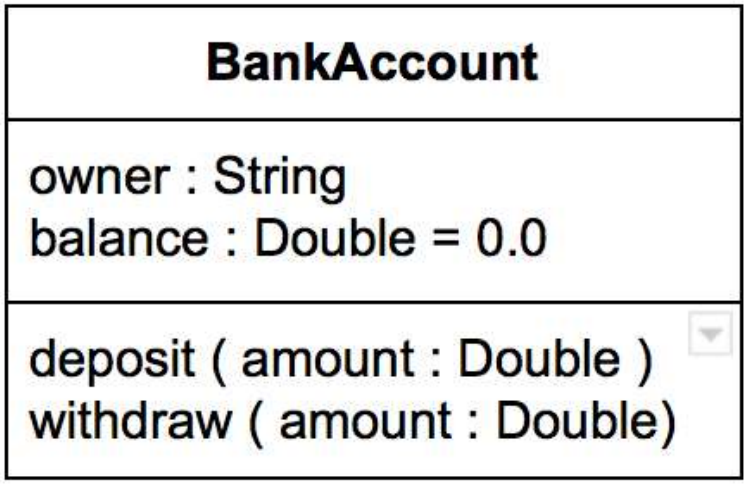
A Class

Una clase se representa como una caja con 3 compartimentos.

El **superior** contiene el nombre de la clase.

El del **medio** contiene los atributos de la clase.

El **inferior** contiene los métodos de la clase..



Podremos utilizar el diagrama de clases para describir el problema y la solución con el patrón.

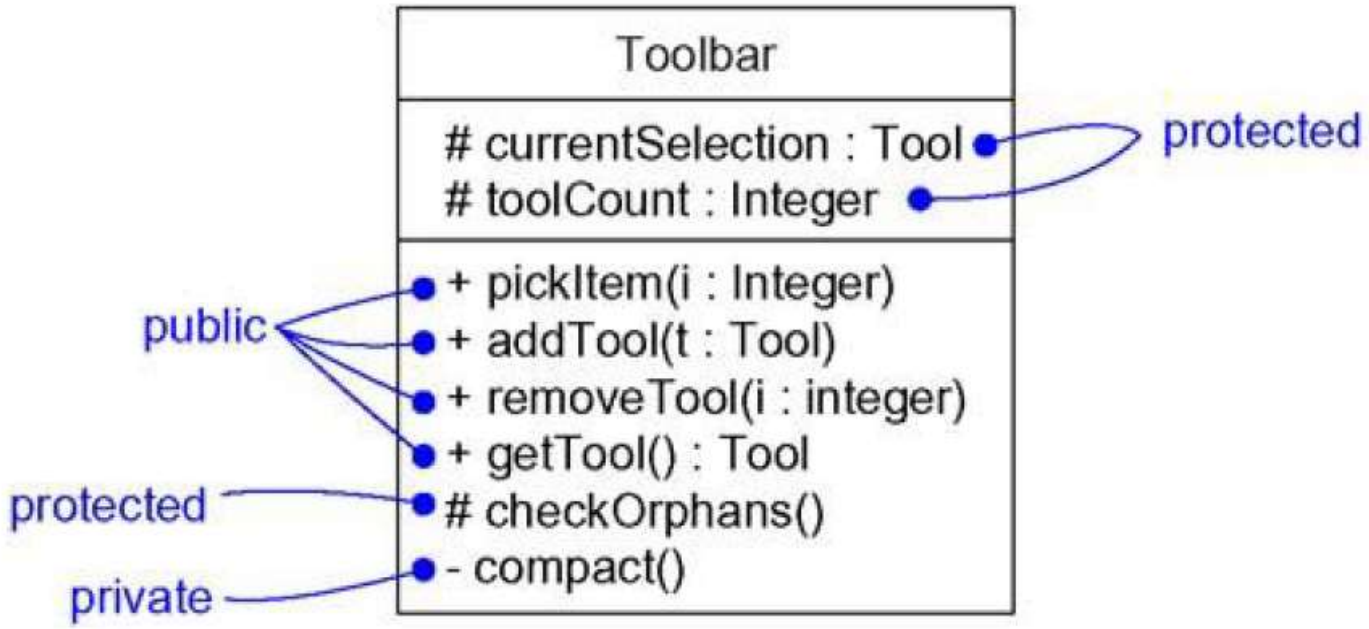
Visibilidad

La definición de un atributo debe especificar que otros objetos los pueden ver. La visibilidad puede ser:

Public (+) permite el acceso a objetos de las otras clases.

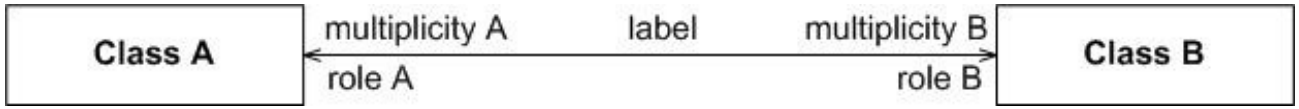
Private (-) limita el acceso a la clase, solo operaciones de la clase tienen acceso.

Protected (#) per
(herencia), las sul
superclase, sino r



Relaciones

Los diagramas de clases pueden contener las siguientes relaciones:



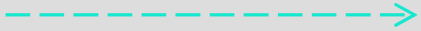
Asociación

Agregación

Composición

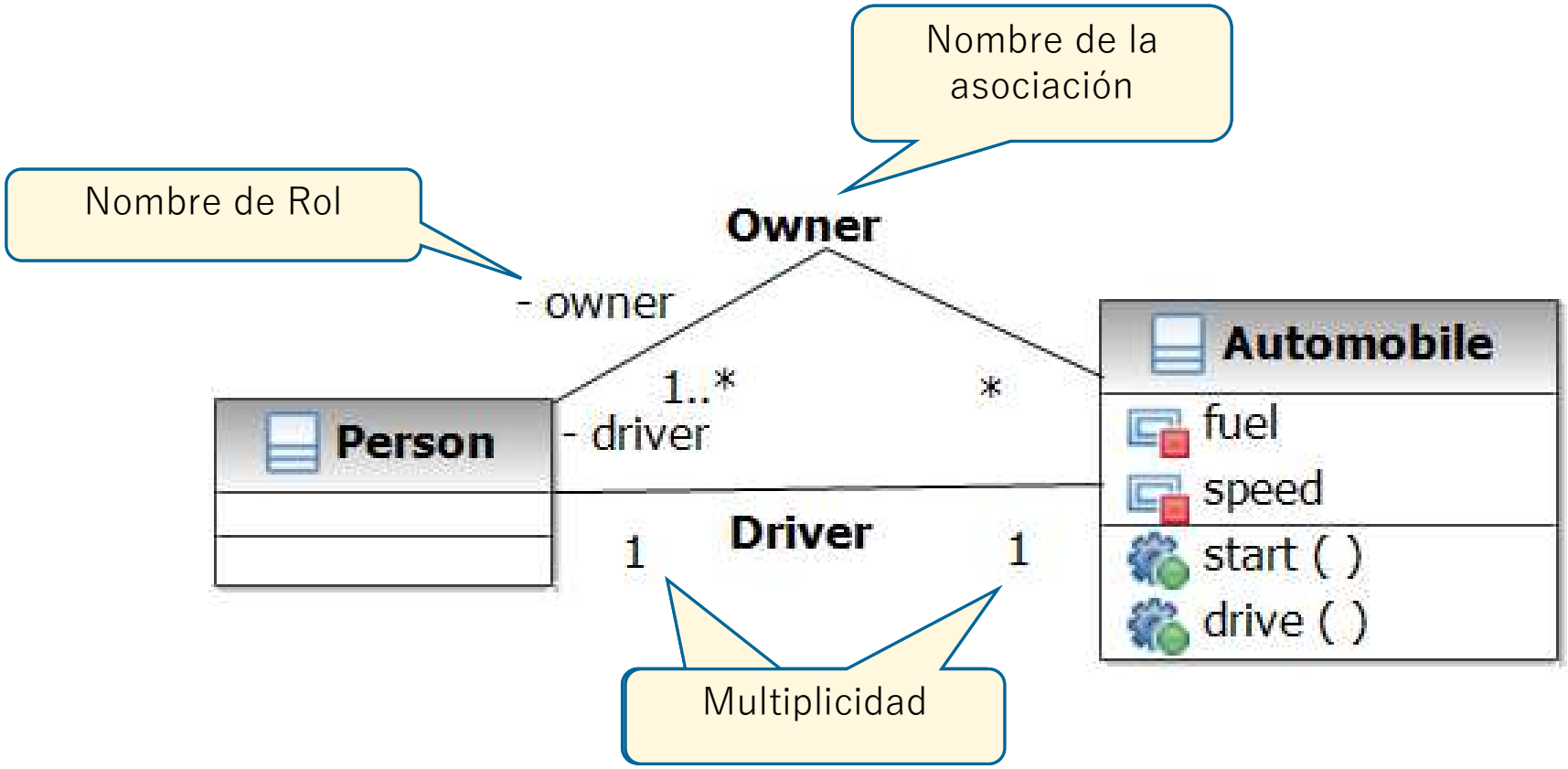
Generalización

Dependencia



Realización

Asociación

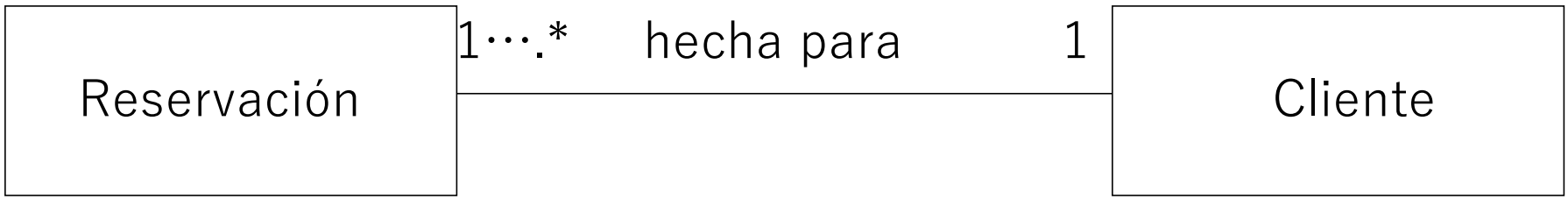

Nombre de la asociación
Nombre de Rol
Owner
- owner
1..*
*
Person
- driver
Automobile
fuel
speed
start ( )
drive ( )
1
Driver
1
Multiplicidad

Dirección

La dirección en las flechas de la asociación, determinan en que dirección puede recorrerse una asociación en el momento de la ejecución.

Una asociación sin flechas significa que se puede ir de un objeto a otro y viceversa.

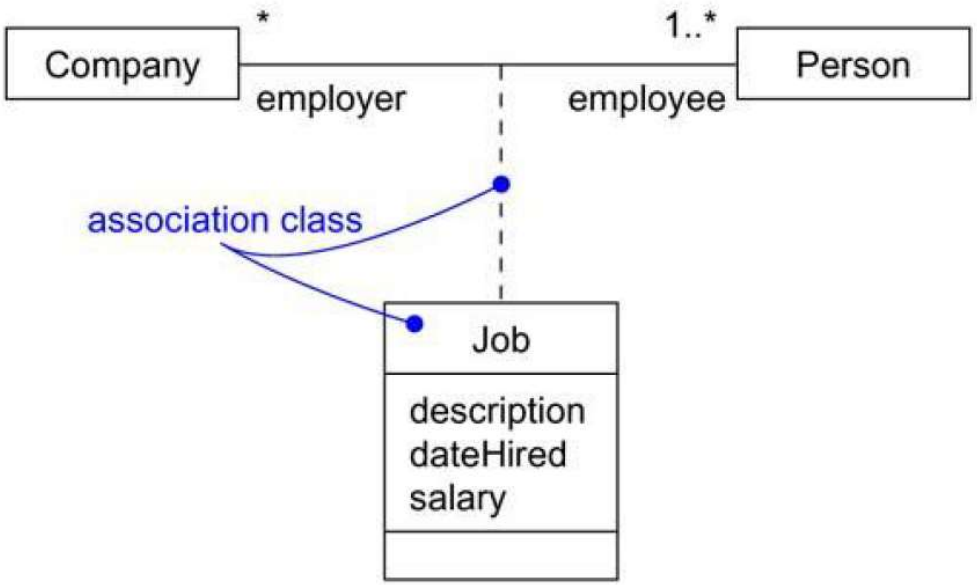
Por ejemplo, la asociación ilustrada implica que desde el objeto Reservación puedes recuperar (dirigirte hacia) el objeto Cliente. También implica que del objeto Cliente puedes recuperar el juego de reservaciones para ese cliente.



Class Association

Cuando se modela una asociación entre clases, a veces es necesario incluir otra clase que contiene información valiosa acerca de la relación.

Se representa como una clase normal solo que la línea que la une con la línea que conecta las asociaciones primarias es punteada.



Ejercicio

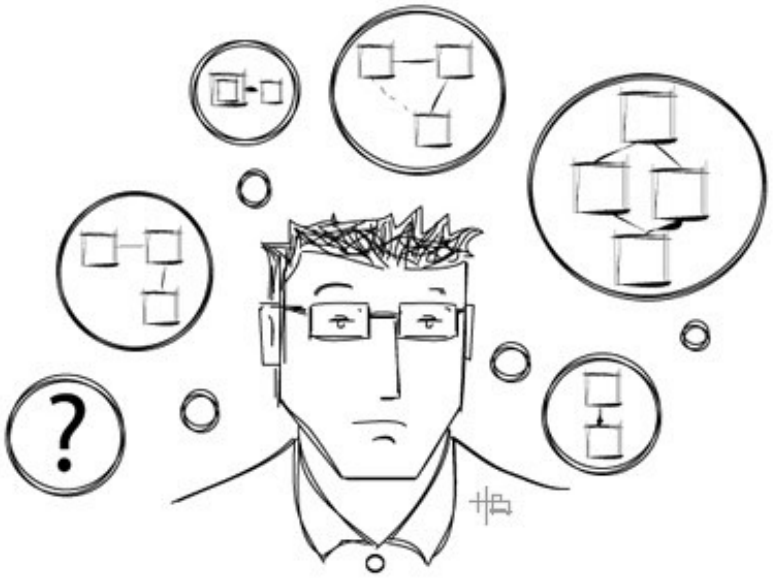
- Los aviones pueden tener uso militar o comercial.
- Los aviones comerciales pueden ser usados como carga o transporte de pasajeros.
- Cada avión está formado por un máximo de 4 motores y de cada motor se conoce la potencia y el rendimiento.
- Los vuelos se identifican por su número, hora de salida, hora de llegada, cuidad de origen y ciudad de destino y se le asigna un avión para su ejecución.
- Un avión puede estar fuera de servicio o atender varios vuelos.
- Se cuenta con un staff de pilotos que se asignan de a dos a cada vuelo según su disponibilidad. De cada piloto se conoce el nombre y las horas de vuelo acumuladas.
- Al final del vuelo cada piloto reporta las irregularidades que ocurrieron durante su tiempo de pilotaje. De cada irregularidad registra la fecha y hora en que ésta sucedió, así como una breve descripción.
- De los aviones de pasajeros se conoce el número máximo de asientos, la cantidad de pasillos, así como si cuenta con sección de primera clase.
- La línea cuenta con un listado de las ciudades del mundo en las que tiene operaciones. Dicho listado está formado por el nombre de la ciudad, temperatura promedio y tipo de clima.
- De cada avión se conoce el modelo y la velocidad y altura máximas de vuelo.

AGENDA

CLASS DIAGRAMS

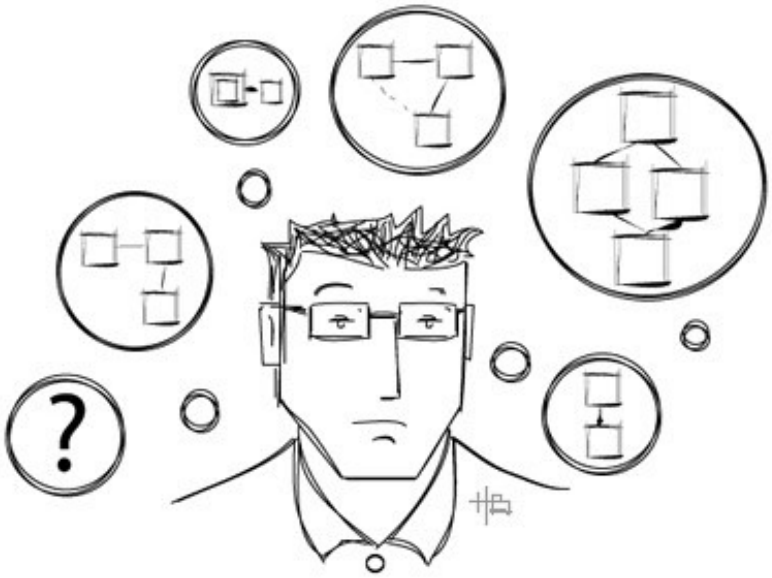
REFLEXIONES

¿En qué nos ayudan los diagramas de clase?



Como veremos

Un patrón es una pieza reutilizable que se produce comúnmente en el sistema de software que proporciona un cierto conjunto de funcionalidades.



La identificación de un patrón también se basa en el contexto en el que se usa.

Por lo tanto, el uso de patrones en el modelado de sistemas ayuda a mantener el diseño estandarizado y, lo que es más importante, minimiza la reinvención de la rueda en el diseño del sistema.

Entonces podrías preguntarte:

¿Cómo se relaciona un patrón con el UML?

Tratar el tema de Design Patterns nos ayudará para entender esto

PREGRADO

Ingeniería de Software

Escuela de Ingeniería de Sistemas y Computación | Facultad de Ingeniería

UPC
Universidad Peruana
de Ciencias Aplicadas

Prolongación Primavera 2390,
Monterrico, Santiago de Surco
Lima 33 - Perú
T 511 313 3333
https://www.upc.edu.pe

exígete, innova