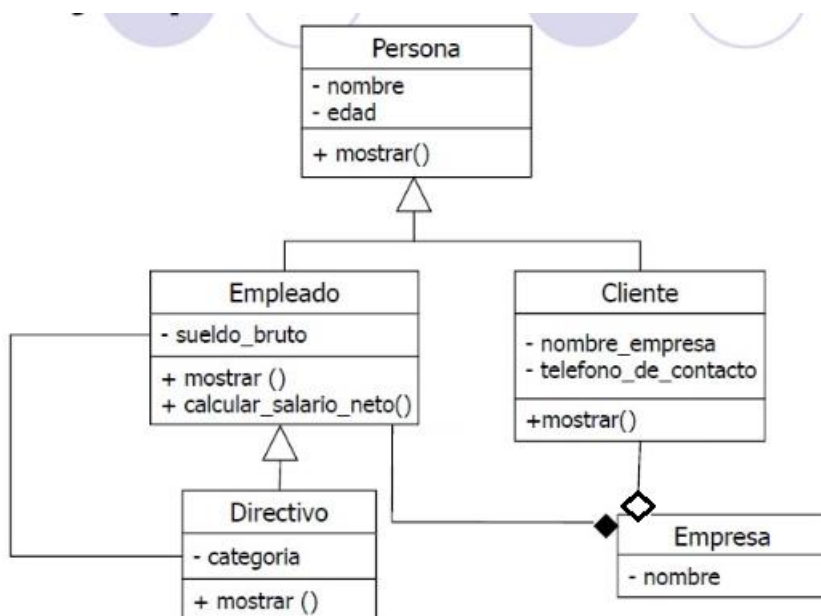


Encuesta para redactar diagramas de clases

Dato el diagrama de clases adjunto a continuación. ¿Cómo lo leerías?, usando un lenguaje coloquial.



Algunas respuestas interesantes

Se describen 3 respuestas textuales de alumnos y docentes anónimos que contribuyeron con el armando.

1) Se ven 5 clases: Persona, Empleado, Cliente, Directivo y Empresa.

Las clases Empleado y Cliente heredan de la clase Persona los atributos nombre y edad, y el método mostrar.

La clase Empleado a parte tiene el atributo Sueldo_Bruto y el método Calcular_salario_Neto(). Un empleado puede ser Directivo y tener empleados a su cargo. La clase Directivo hereda de empleado, y tiene el atributo Categoría.

La clase cliente tiene como atributos Nombre_Empresa y Telefono_de_Contacto.

Y la clase Empresa tiene el atributo nombre.

La clase empresa está compuesta por Empleados. Es decir que, si no hay empleados, no existe la empresa. Es una relación de composición.

Y la relación que hay entre empresa y cliente es de agregación. Es decir que la empresa agrupa un grupo de clientes. Una relación de uno a muchos.

2) Tengo una clase Persona, con atributos privados nombre de tipo String y edad de tipo entero, que tiene un método publico mostrar, que no recibe parámetros ni devuelve nada.

Después tengo Empleado, que hereda de Persona y compone a la clase Empresa, y tiene un atributo privado sueldo_bruto de tipo doble, y métodos públicos mostrar que no recibe parámetros ni devuelve nada y calcular_sueldo_netto que tampoco parámetros y devuelve un doble.

También tengo una clase Cliente que hereda de Persona y tiene una asociación con Empresa, que tiene atributos privados llamados nombre_empresa de tipo String y teléfono_de_contacto de tipo String, y un método publico llamado mostrar, que no recibe parámetros ni devuelve nada

Además, tengo una clase Directivo que hereda de Empleado y esta a la vez relacionado con el, con atributo privado categoría de tipo entero, y un método publico mostrar que no devuelve nada y no tiene parámetros.

Por último, tengo una clase Empresa, que tiene un atributo privado nombre de tipo String.

3) (yo en tu lugar tomara como base el de más arriba para partir de algo) la superclase Persona con el atributo nombre tipo privado y el atributo edad tipo privado (acá todos los atributo y tipo q sean necesarios) además (la palabra además o una palabra especial para separar q terminaron los atributos y empiezan los metodos) los métodos mostrar tipo publico sin variables entrantes hereda a dos clases: Empleado y Cliente (acá dejaria una pequeña pausa) la superclase Empleado con el atributo sueldo_bruto tipo privado además con los metodos tipo publico mostrar sin variables entrantes y el metodo tipo publico calcular_salario_netto sin variables entrantes hereda a la clase Directivo y tiene una relacion de composicion con la clase empresa la clase Cliente con los atributos nombre_empresa tipo privado y el atributo telefono_de_contacto tipo privado además con el metodo mostrar tipo publico sin variables entrantes tiene una relacion de agregacion con la clase empresa la clase Directivo (decidi seguir por esta clase xq cuando fui a empleado primero me quedo en el tintero (por así decirlo) directivo entonces es una clase q mencione antes q empresa) con el atributo categoria tipo privado además con el metodo mostrar tipo publico sin variables entrantes la clase Empresa con el atributo nombre tipo privado además no posee métodos.

Conclusión

En base a estas tres respuestas y otras 4 más que me parecían bien ejemplificadas por estas tres, se puede sacar la conclusión que el orden en el que se describen las clases no es primordial, pero si lo es el orden de su estructura interna, se ve que definen primero sus atributos, luego sus métodos y por último sus relaciones. También ha salido a la luz que en caso de que no tenga atributos, métodos o relaciones es bueno remarcarlo para que no parezca que es un dictado incompleto. Es fundamental realizar pausas entre las clases para definir rupturas en la lectura que describan mejor la lógica.