

Caso de estudio – Clase 4

Utilización de Composición, colecciones, herencia y polimorfismo

Enunciado

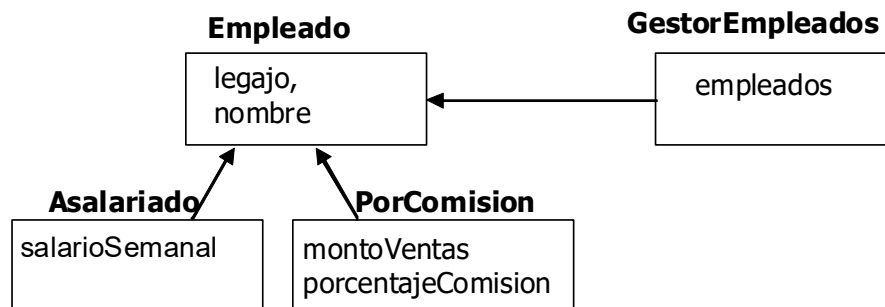
El departamento de recursos humanos, de una importante empresa nos ha solicitado el desarrollo de un programa, con el propósito de gestionar información relacionada con el salario que perciben sus empleados.

En la etapa de análisis se han detectado las siguientes entidades:

Empleado: que tiene los siguientes datos: legajo y nombre.

Empleado Asalariado: (es un Empleado) que tiene como atributo el monto del salario semanal. El cálculo del ingreso de este empleado se establece por el monto del salario semanal.

Empleado por Comisión: (es un Empleado) que tiene como datos: el monto de ventas por semana y el porcentaje de comisión. El ingreso para estos empleados se calcula como $\text{porcentajeComision} * \text{montoVentas} / 100$.



Se pide:

1. Implementar las clases Empleado, Asalariado y PorComision que tengan los métodos para inicializar, los métodos de acceso y modificadores, el método asString, que retorna una cadena con los atributos y el método montoIngreso, que retorna el ingreso que le corresponde según lo enunciado anteriormente.
2. Implementar la clase GestorEmpleados que maneje una colección de empleados. Los metodos solicitados son:
 - a. Inicializador: Crea la coleccion
 - b. add: Agrega un objeto a la colección, verificando que no se repita el legajo de un empleado.
 - c. asString: Retorna un String con todos los empleados registrados.
 - d. montoTotalAPagar: Retorna el importe total que la empresa tiene que pagar en concepto de ingresos a sus empleados.
 - e. cantidadAsalariados: Retorna la cantidad de empleados asalariados que tiene un ingreso superior a un valor "x" recibido como colaborador externo.

- f. rangoMontoVentas: Retornar los datos de los empleados que tengan un monto de ventas entre dos valores recibidos como colaboradores externos.
 - g. listadoOrdenado: Retornar el listado de empleados ordenados por legajo de mayor a menor.
 - h. buscarEmpleado: Retornar los datos del empleado que tenga un legajo determinado recibido como colaborador externo, si no lo encuentra informar
3. Probar con objetos simples o implementar una clase Aplicación que instancie la clase GestorEmpleados e invoque sus métodos.