Control Nº5

**ArrayList**

**Nombre : Fecha: 6 de junio 2016**

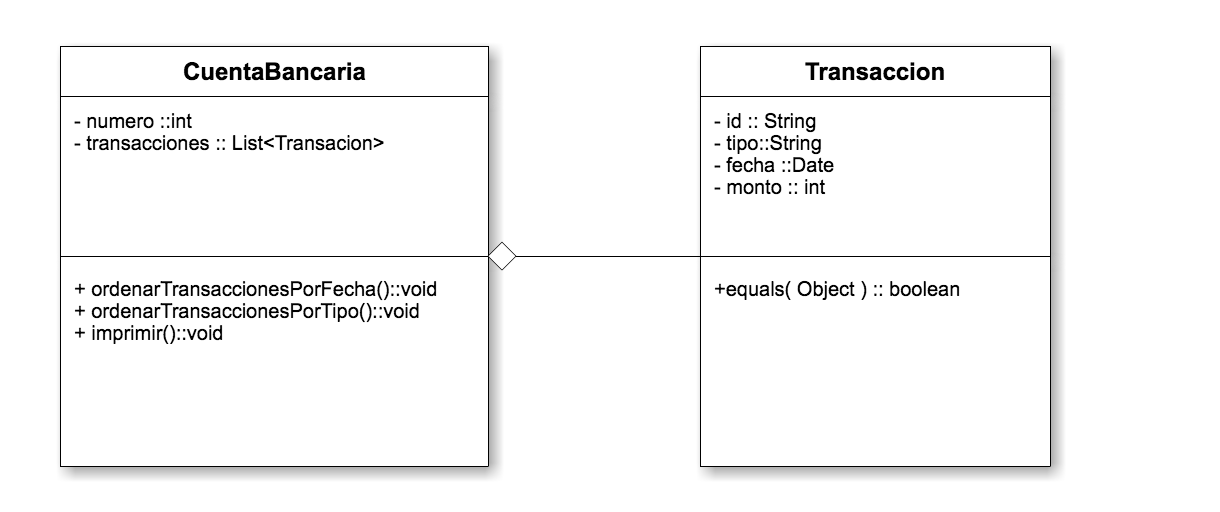
Considere lo siguiente: cada objeto debe encapsular sus atributos, generar getters y setters para cada uno de ellos. Crear un constructor por defecto que inicialice los atributos de la clase y un constructor que reciba como parámetro todos los atributos de la clase.

**Nombre del package cl.curso.java.control\_cinco.jmaldonado**

Reemplazar jmaldonado por la inicial de su nombre y su apellido.

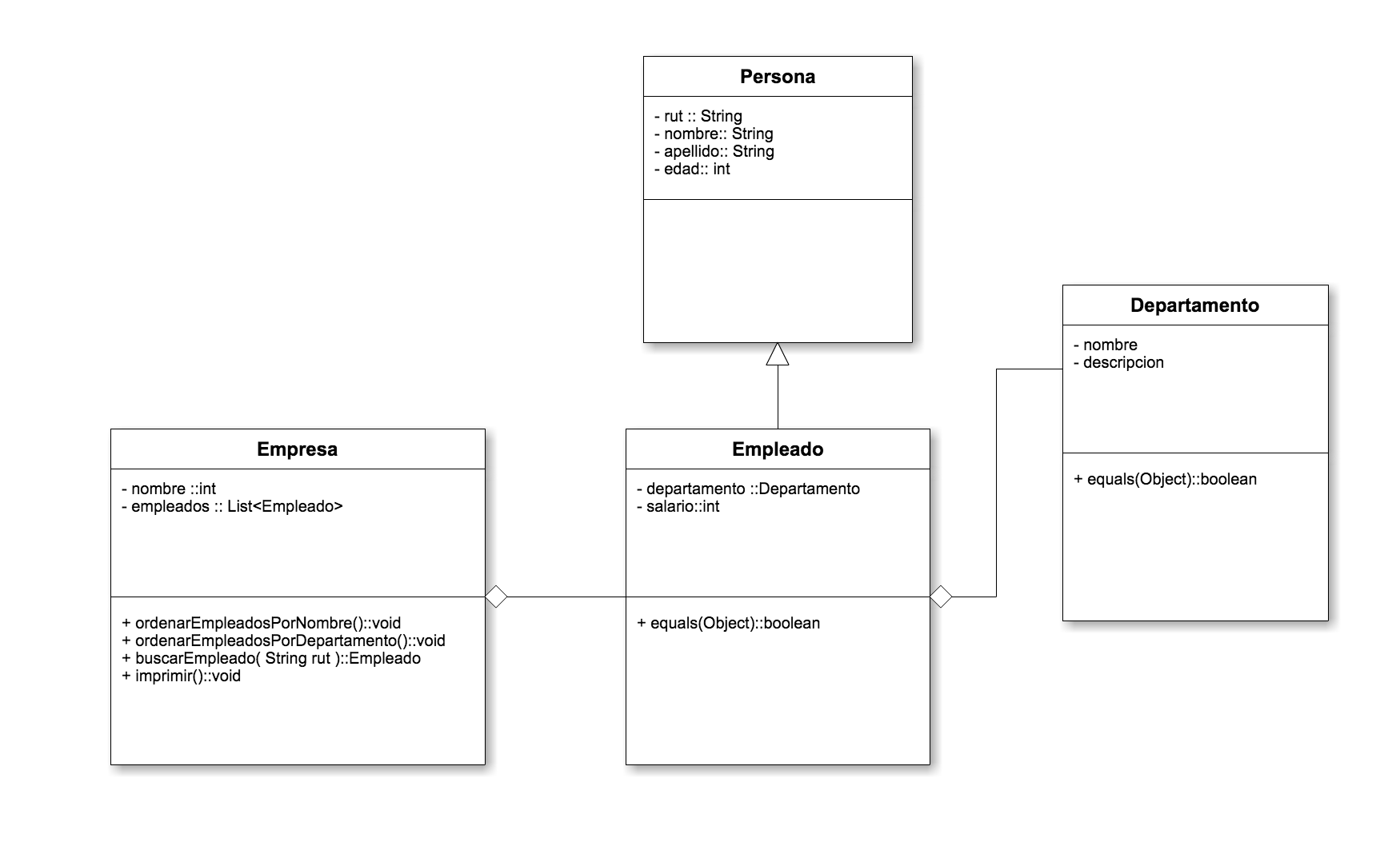
En los ejercicios prácticos se evaluará lo siguiente

1. Según el siguiente diagrama, escribir las clases necesarias.



* El tipo de transacción puede ser “deposito” o “abono”.
* El método ordenarTransaccionesPorFecha()::void debe ordenar la lista de transacciones por la fecha. (1 pts.)
* El método ordenarTransaccionesPorTipo()::void, debe ordenar la lista de transacciones por el tipo, debe mostrar primero las de deposito y luego las de abono. (1 pts.)
* El método imprimir()::void debe imprimir la lista de transacciones usando un ciclo Foreach (1 pts.)
* Uso de buenas practicas al escribir métodos, variables, clases e interfaces. (1 pts.)
* Identación del código. (1 pts.)
* Creación de Constructores (1 pts.)
* Encapsulación getters y setters (1pts)
* Uso de comentarios (1pts).

1. Según el siguiente diagrama, escribir las clases necesarias.



* El método ordenarEmpleadosPorNombre()::void debe ordenar la lista de empleados por nombre de empleado (1 pts.)
* El método ordenarEmpleadosPorDepartamento()::void debe ordenar la lista de empleados por el departamento al que pertenece. (1 pts.)
* El método buscarEmpleado( String rut )::Empleado, buscara un empleado por su rut, el método retornara un Empleado, si no encuentra el empleado dentro de la lista, lanzara una excepción no checkeada definida por usted. (1 pts.)
* El método imprimir()::void debe imprimir la lista de empleados utilizando un ciclo while con iterator.
* La clase Empleado debe sobrescribir el método equals( Object ), el cual comparara si dos empleados son iguales según su rut.
* Uso de buenas practicas al escribir métodos, variables, clases e interfaces. (1 pts.)
* Identación del código. (1 pts.)
* Creación de Constructores (1 pts.)
* Encapsulación getters y setters (1pts)
* Uso de comentarios (1pts).