**Práctica # 8 Herencia II**

Nombre: GIAN MOLINA RAIGOZA Matrícula:1636155 Calificación: \_\_\_\_\_\_ Ponderación: 11% Día: \_\_\_\_\_\_\_\_\_\_\_\_ Hora: \_\_\_\_\_\_: \_\_\_\_\_\_

**Objetivos Generales:**

Entender y aplicar el concepto de herencia y el uso de clases y métodos abstractos.

**Actividad #1**

Escribir un programa que funcione como un administrador de una compañía comercializadora de materias primas:

1. Crear cuatro archivos:
   * ***Administrador.java*** que contendrá el método **main**.
   * ***Entidad.java*** que contendrá la definición de la clase padre abstracta.
   * ***Empleado.java*** que contendrá la definición de un Empleado heredando de la clase entidad.
   * ***Proveedor.java*** que contendrá la definición de un Proveedor heredando de la clase entidad.

**En el archivo *Entidad.java*:**

1. Crear una clase abstracta con los atributos y métodos especificados en el diagrama de clases.
   1. Los métodos *imprimir()* y *getCodigo()* deben de ser declarados como abstract.
      1. El método imprimir será implementado en las clases hijas para imprimir en consola la información del objeto.
      2. El método código generara un código para cada una de las clases hijas, el funcionamiento de este método se explica mas a detalle a continuación.

**Para los archivos *Empleado.java* y *Proveedor.java*:**

1. Heredar la clase ***Entidad.java***
2. Programar los atributos y métodos especificados en el diagrama de clases.
3. Implementar los métodos abstractos definidos en la clase padre.
   1. Para el archivo ***Empleado.java****.*
      1. El método getCodigo() generara un string que este compuesto por las cuatro primeras letras del Nombre y su departamento.
   2. Para el archivo ***Proveedor.java***.
      1. El método getCodigo() generara un string que este compuesto por las cuatro primeras letras del Nombre y su producto.

**En el archivo *Administrador.java*:**

1. Declarar 2 arreglos, uno de tipo ***Empleado*** y otro de tipo ***Proveedor***, cada uno de 5 elementos.
2. Declarar un ciclo que tenga las siguientes opciones:
   1. Agregar un Empleado.
      1. Se agregará un Empleado al arreglo de tipo Empleado.
      2. Solo se podrán agregar 5 empleados.
   2. Agregar un Proveedor.
      1. Se agregará un Proveedor al arreglo de tipo Proveedor.
      2. Solo se podrán agregar 5 proveedores.
   3. Imprimir Empleado
      1. Se le pedirá al usuario el código del Empleado y el programa mostrará en consola la información del empleado con ese código.
   4. Imprimir Proveedor
      1. Se le pedirá al usuario el código del Proveedor y el programa mostrará en consola la información del proveedor con ese código.
   5. Salir.

|  |
| --- |
| **Entidad** |
| - String nombre; //Nombre del objeto  - String tipo; //Tipo del objeto (Empleado o Proveedor) |
| + abstract void imprimir();  + abstract String getCodigo();  + get y set de los atributos; |

|  |
| --- |
| **Empleado** extends **Entidad** |
| - double sueldo;  - String departamento; |
| + get y set de los atributos  + definiciones de los métodos abstractos de Entidad |

|  |
| --- |
| **Proveedor** extends **Entidad** |
| - int cantProducto;  - double precioProducto;  - String tipoProducto; |
| + get y set de los atributos  + definiciones de los métodos abstractos de Entidad |