**FACULTAD DE INGENIERÍA**

**MATERIA: PROGRAMACIÓN ORIENTADA A OBJETOS**

**EXAMEN FINAL**

|  |  |
| --- | --- |
| Nombre: | Profesor: DRA. MARÍA VICTORIA CARRERAS CRUZ |
| Licenciatura: | Calificación: |
| Fecha: 9 DE JULIO DEL 2021 | Firma: |

*“Certifico que el trabajo realizado en este examen es estrictamente personal y reconozco que no cumplir las reglas o COMETER ACTOS DESHONESTOS en los exámenes puede resultar en la baja definitiva de la Universidad”.*

**INSTRUCCIONES GENERALES:**

1.- Programar en Python el sistema descrito a continuación y subir a Moodle al término del examen un archivo en Word. Dicho archivo debe contener el código generado por Ud. y la impresión de las pantallas necesarias.

2.- El archivo debe llamarse con su nombre completo.

3.- No puede registrarse ningún acceso a Internet excepto el acceso para subir el examen.

4.- Una vez culminado el tiempo reglamentario de examen el acceso a Moodle se cerrará sin ninguna posibilidad de que el profesor u otra persona lo pueda volver a abrir. Por lo que te recomendamos subir tu examen a más tardar 10 minutos antes del horario estipulado de fin de examen. Por ningún motivo se aceptarán archivos extemporáneos.

5.- El examen es individual.

**EXAMEN**

1. El problema que se desea solucionar se encuentra en el entorno de una Tienda Departamental. Genere una clase llamada **Dependiente** que siga las siguientes condiciones:

* Sus atributos son: **nombre, edad, sexo**(H hombre, M mujer)**, idEmpleado.**
* No se desea que se acceda directamente a la edad. Si no que sea a través de una propiedad y sus métodos correspondientes. La edad capturada no podrá ser menor a 18 años. Si es menor a eso se le asignará el valor de 18.
* El **idEmpleado** debe ser privado y solamente se podrá llenar a través del método **llenarID().**
* Los métodos que se implementarán son:
  + **cobrar ()** Recibe como parámetro de entrada el precio del producto y el pago del cliente. Si es suficiente emite un cartel que diga “El cliente pagó (pago) y hay que devolverle (pago – precio)”, de lo contrario dice “No es suficiente lo cobrado para pagar el producto”.
  + **darRecibo()** Indica que se entregó el recibo a través de un cartel.
  + **mostrarDatos():** Mostrar toda la información del objeto.

1. Crear una subclase que herede de **Dependiente** y que sea **Vendedor**.

* Debe tener todos los atributos y métodos de la clase **Dependiente**, pero además debe tener el atributo **dineroVentas** y este debe llenarse también a través del constructor.
* El método **cobrar()**, aparte de hacer lo mismo que en la clase **Dependiente**, debe acumular lo que va cobrando.

1. Crear una subclase que herede de **Dependiente** y que sea **Emisor\_de\_Facturas**.

. Debe tener todos los atributos y métodos de la clase **Dependiente**, pero además debe tener el atributo **pagoFactura** y este debe llenarse también a través del constructor.

. El método **cobrar()**, aparte de hacer lo mismo que en la clase **Dependiente**, debe acumular lo que va cobrando y sumarle el 16 %.

1. Crear una subclase **Vendedor\_que\_factura** que herede de **Vendedor** y de **Emisor\_de\_Facturas** y que en **pago\_factura** ponga por default 1,000.

Su método **cobrar()** debe ser el mismo de la clase **Emisor\_de\_Facturas**.

1. La clase **Vendedor\_que\_factura** debe tener un método **facturas\_especiales()** que simule la sobrecarga de métodos. Este método debe tener uno o dos parámetros de entrada: **facturacion\_especial** que es un número flotante que entra como parámetro de entrada y que está entre 20,000 y 100,000 y **pago\_extra** que puede ser definido por default con un 0 o ser capturado. La salida de este método es “La facturación especial de (nombre) es (facturación\_especial + pago\_extra)”.
2. La clase **Vendedor\_que\_factura** debe tener un destructor que emita un cartel que diga “Este vendedor que factura fue eliminado”.
3. Deben manejarse todas las excepciones que correspondan para cumplimentar la calidad de código.
4. Crear tres objetos de la tres subclases que muestren **TODA** la funcionalidad del código.
5. Mencione una ventaja de utilizar el paradigma de Programación Funcional y explíquela brevemente.