Cuestionario .NET

1) La situación es esta: tú trabajas en una empresa donde los vendedores reciben comisiones del 20% por sus ventas totales, y tu jefe quiere que ayudes a los vendedores a calcular sus comisiones creando un programa que les pregunte su nombre y cuánto han vendido en este mes. Tu programa le va a responder con una frase que incluya su nombre y el monto que le corresponde por las comisiones. Recuerda que los ingresos de usuarios se almacenan como strings. Por lo tanto, deberías convertir uno de esos ingresos en un float para poder hacer operaciones con él.

2) El programa va a elegir una palabra secreta y le va a mostrar al jugador solamente una serie de guiones que representa la cantidad de letras que tiene la palabra. El jugador en cada turno deberá elegir una letra y si la letra se encuentra en la palabra oculta, el sistema le va a mostrar en qué lugares se encuentra. Pero si el jugador dice una letra que no se encuentra en la palabra oculta, pierde una vida.

3) Entonces, antes de comenzar, es necesario que crees en tu ordenador un directorio en la carpeta base de tu ordenador, con una carpeta llamada Kosmos, que contiene cuatro carpetas ( categorías de informes) y cada una de ellas contiene dos archivos de texto que son informes. Dentro de los archivos puedes escribir lo que quieras. Lo importante es que escribas algo para poder leerlas cuando haga falta o, si prefieres, también puedes directamente descargar y descomprimir el archivo adjunto a esta elección y ubicarlo en tu directorio raíz si no tienes ganas de crearlo tú mismo. Aquí viene la consigna: tu código le va a dar primero la bienvenida al usuario, le va a informar la ruta de acceso al directorio donde se encuentra nuestra carpeta de informes, le va a informar cuántos informes hay en total dentro de esa carpeta, y luego le va a pedir que elija una de estas opciones que tenemos aquí:

La opción 1 le va a preguntar qué categoría elige (Marketing, Ventas, Compras, métricas), y una vez que el usuario elija una, le va a preguntar qué informe quiere leer, y mostrar su contenido. 2. En la opción 2 también se le va a hacer elegir una categoría, pero luego le va a pedir que escriba el nombre y el contenido del nuevo informe que quiere crear, y el programa va a crear ese archivo en el lugar correcto. La opción 3 le va a preguntar el nombre de la categoría que quiere crear y va a generar una carpeta nueva con ese nombre. 4. La opción 4, hará todo lo mismo que la opción uno, pero en vez de leer el informe, lo va a eliminar 5. La opción 5, le va a preguntar qué categoría quiere eliminar 6. Finalmente, la opción 6 simplemente va a finalizar la ejecución del código.

4)Programación OO. **Definición de Clases y Métodos:**

* + Define una clase Empleado que tenga atributos como nombre, apellido, salario y un método para calcular el salario anual.
  + Crea una instancia de la clase Empleado y muestra su salario anual.

1. **Herencia:**
   * Crea una clase Gerente que herede de la clase Empleado y tenga un atributo adicional departamento.
   * Agrega un método a la clase Gerente para incrementar su salario en un cierto porcentaje.
2. **Encapsulación y Propiedades:**
   * Modifica la clase Empleado para hacer que el salario sea un atributo privado y proporcione métodos getter y setter para acceder y modificar el salario.

5) Crea una web con una lista de tareas de este año , crea tabla de tareas, buscador de tareas y creación de tareas, trata de usar los colores de Konecta ( gamas de azules) , agrega los crud relacionados.

6) Bonus Api Rest:

**Título del Ejercicio:** Aplicación de Clima utilizando una API REST en .NET

**Descripción:** Desarrolla una aplicación en VB .Net que utilice una API REST para obtener datos climáticos de una ciudad específica y muestre estos datos de manera legible para el usuario. Puedes utilizar la API de OpenWeatherMap (https://openweathermap.org/api) u otra de tu elección que proporcione datos climáticos.

**Requisitos Técnicos:**

1. Utilizar VB .Net como lenguaje de programación.
2. Consumir datos de una API REST para obtener información climática en tiempo real.
3. Implementar una interfaz de usuario simple que muestre la información obtenida de la API de manera legible y organizada.
4. Permitir al usuario ingresar el nombre de la ciudad para la cual desea conocer el clima.
5. Mostrar al menos los siguientes datos: temperatura actual, descripción del clima, y sensación térmica.
6. Manejar posibles errores de conexión con la API y errores de entrada del usuario.

**Bonus:**

* Permitir al usuario elegir entre mostrar la temperatura en Celsius o Fahrenheit.
* Agregar funcionalidades adicionales, como pronóstico extendido, gráficos de temperatura, etc.
* Implementar pruebas unitarias para verificar el correcto funcionamiento del código.

**Entregables:**

* Código fuente de la aplicación en VB.NET.
* Documentación que explique cómo ejecutar la aplicación y cualquier consideración adicional.

**Observaciones:** Recuerda seguir las mejores prácticas de programación, como el modularidad, el manejo adecuado de errores y la documentación del código.