



Prueba de desempeño del módulo C#

Team Leader: Javier Combita

Clan: Banners Lee

Reglas de la prueba

Comunicación: Está prohibido hablar o comunicarse de cualquier forma con otros coders durante el examen.

Integridad Académica: Cualquier forma de trampa, incluido el plagio, copia o uso de material no autorizado (autogeneradores de código o inteligencia artificial), resultará en una calificación de cero en la prueba y puede llevar a sanciones adicionales según las políticas de RIWI.

Idioma: Todo el código debe ser 100% en inglés, lo único que se permite en español son los comentarios de ayuda, los datos en sí y la interfaz visual de la prueba

Tecnologías Permitidas

- CLI de .NET
- Visual Studio Code
- Git
- GitHub

Material Permitido: Solo se permite ver material de apoyo como lo son diapositivas o ejercicios realizados en clase y notas tomadas durante las sesiones de entrenamiento, ya sea en el cuaderno o de forma digital, acordando con el TL cuáles son los recursos que va a usar, previo inicio de la jornada.

Están permitidas las siguientes páginas para acceso:

- [Guía de C#: lenguaje administrado de .NET | Microsoft Learn](#)

- [GitHub - Riwi-io-Medellin/C-Sharp_Fundamentals](#)

Permanencia en el Aula:



Una vez iniciada la prueba, no se permite salir del aula hasta haber entregado el examen, salvo en los siguientes casos:

- **Descansos:** Se permite un descanso de 20 minutos cada 3 horas.
- **Salidas al baño:** Los coders pueden salir al baño cuando sea necesario.
- **Eventualidades:** En caso de alguna eventualidad, se permitirá salir del aula.

Entrega:

Una vez finalizada la prueba, se debe subir una carpeta comprimida a la plataforma Moodle. La carpeta comprimida debe nombrarse como **PruebaC-sharp_NombreApellido** del coder y debe contener:

- La carpeta con el contenido del proyecto en formato .zip.
- Un archivo .txt con el enlace al repositorio de GitHub de la prueba. El repositorio debe estar público.

La hora máxima de entrega es a las **14:00 horas**. Las pruebas enviadas después de esta hora **NO** serán consideradas.

Medio de Entrega:

El único medio para la entrega es la plataforma **Moodle**. Se recomienda enviar la prueba 10 minutos antes de la hora límite en caso de eventualidades.

Veterinary Center

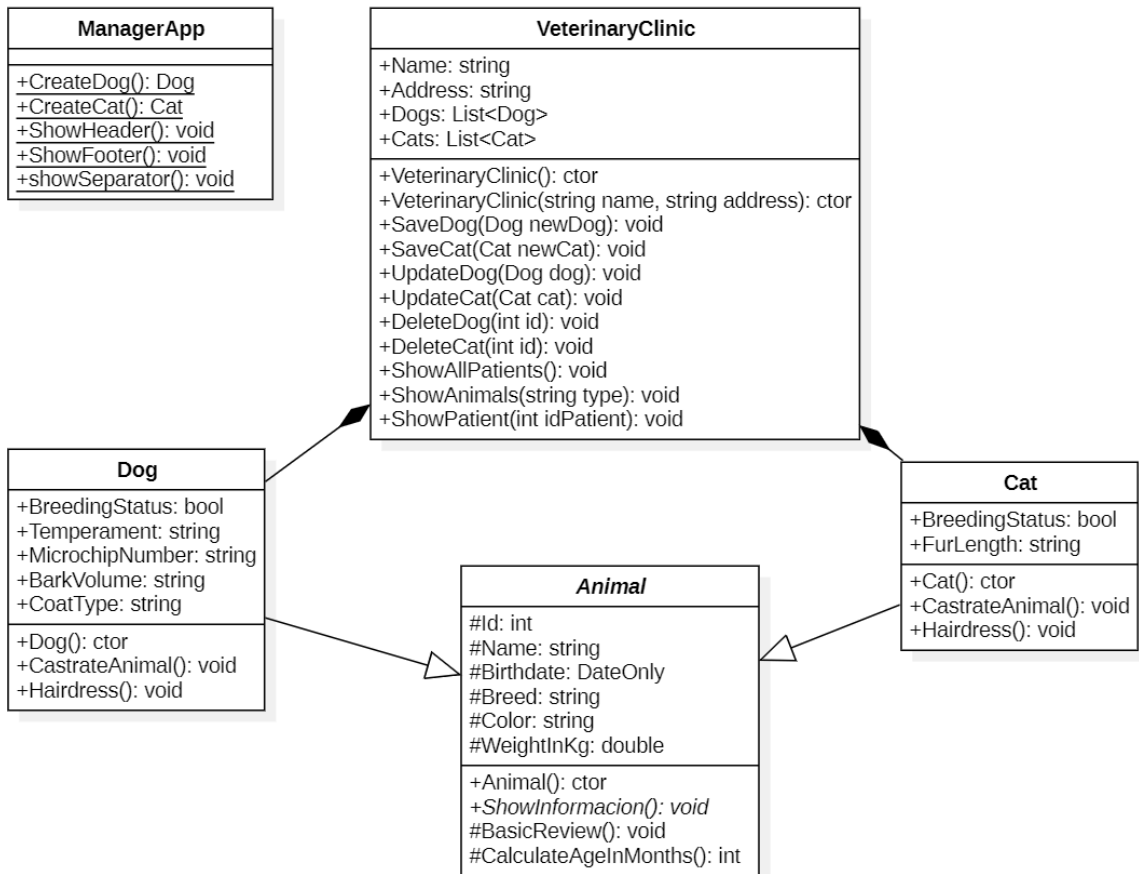


Has sido contratado por una reconocida clínica veterinaria para desarrollar un sistema que les permita gestionar de manera eficiente los registros de sus pacientes, los cuales incluyen tanto perros como gatos. Este sistema será utilizado por el personal veterinario para mantener actualizada la información de los animales, así como para registrar y consultar datos relevantes sobre ellos.

Objetivos de la prueba

El objetivo de esta prueba es que demuestres tus habilidades en el diseño e implementación de un sistema orientado a objetos utilizando C#. Deberás desarrollar las clases y métodos necesarios siguiendo el diagrama de clases proporcionado y cumplir con las funcionalidades requeridas por la clínica veterinaria.

Diagrama de Clases



Requisitos de Implementación

1. Implementación de Clases y Métodos:

- Implementa las clases y sus métodos según el diagrama proporcionado.
- Asegúrate de que los métodos ShowInformacion en las clases derivadas Cat y Dog muestren la información básica del animal.
- Implementa los métodos BasicReview y CalculateAgeInMonths en la clase Animal.

2. Gestión de la Clínica Veterinaria:



- Implementa las funcionalidades de la clase VeterinaryClinic para guardar, actualizar, eliminar y mostrar la información de los pacientes.
 - Asegúrate de que los métodos ShowAllPatients, ShowAnimals y ShowPatient muestren la información correctamente.
3. **Aplicación de Gestión:**
- Utiliza la clase ManagerApp para crear instancias de Dog y Cat, y para mostrar encabezados, pies de página y separadores en la salida de la consola.
4. **Interfaz visual:**
- Mediante la consola se debe poder administrar la aplicación y realizar las operaciones necesarias.

Reglas y Restricciones Adicionales

1. **Castración de Animales:**
 - Un animal no puede ser castrado más de una vez. Debes asegurarte de implementar la lógica necesaria para verificar esto antes de proceder con la castración.
2. **Temperamentos en Perros:**
 - Los tipos de temperamento permitidos para los perros son: Tímido, Normal y Agresivo. Implementa la lógica necesaria para validar estos valores.
3. **Peluquería de Animales:**
 - No se puede peluquear a un perro si tiene el pelo corto. Asegúrate de que el método Hairdress en la clase Dog verifique la longitud del pelo antes de proceder.
 - No se puede peluquear a un gato que no tiene pelo. Implementa la lógica necesaria en el sistema para manejar esta restricción.
4. **Tipos de Pelo Permitidos:**
 - Los tipos de pelo permitidos para los animales son: Sin pelo, Pelo corto, Pelo mediano, Pelo largo. Implementa la lógica para validar que los atributos FurLength en Cat y CoatType en Dog solo acepten estos valores.
5. **Entradas de Datos:**
 - Asegúrate de validar todas las entradas de datos para los atributos de las clases, incluyendo tipos de datos, rangos de valores y restricciones específicas (por



ejemplo, la fecha de nacimiento no puede ser una fecha futura o un correo no puede quedar con espacios en blanco al comienzo).

Entregables (Moodle)

- Código fuente de las clases y métodos implementados.
- Uso de Git y GitHub para gestionar el código (Link del repo).

Criterios de Evaluación

- Revisar la rúbrica disponible en Moodle (El TL informará donde se puede consultar)

Esta prueba te permitirá demostrar tus competencias en programación Backend con el lenguaje de programación C#.

¡Buena suerte!