Bonus Challenge

He usado Llama 3 y DeepSeek para la investigación y realización de este bonus challenge, donde se arregla el método o el hecho de que en el transcurso del programa podíamos cambiar la propiedad tipo licencia.

Las mejoras que me ayudaron a realizar fueron:

1. Validación en propiedades: Ahora tanto Edad como TipoLicencia validan cuando se modifican.
2. Normalización de datos: Convertimos siempre el tipo de licencia a mayúsculas para evitar problemas con "a" vs "A".
3. Mejores mensajes de error: Los mensajes de excepción ahora incluyen los valores específicos que causaron el error.
4. Mejor estructura del código: Usé un switch en EdadParaLicencia para mayor claridad.
5. Manejo de casos no válidos: Si se introduce un tipo de licencia no reconocido, se lanza una excepción.
6. Método MostrarInformacion: Lo mejoré para que también muestre la edad.
7. Encapsulamiento: Usé campos privados para las propiedades que necesitan validación.

Me puse a trabajar en el código de la clase Chofer y logré implementar la validación para asegurarme de que la edad sea adecuada para el tipo de licencia. La validación se realiza en el constructor y también cuando se cambia la propiedad TipoLicencia, lo que garantiza que la propiedad siempre tenga un valor válido.

En específico, agregué una condición en el setter de la propiedad TipoLicencia que verifica si la edad es adecuada para el tipo de licencia seleccionado. Si no es así, se lanza una excepción para indicar el error.

Me parece que esta implementación es una buena forma de asegurarme de que los datos sean consistentes y válidos, y estoy contento de haber podido resolver este desafío.

Mi código corregido:

namespace p1bpoo.MisClases

{

internal class Chofer

{

public string Nombre { get; set; }

public int Edad { get; private set; }

private string tipoLicencia;

public string TipoLicencia

{

get { return tipoLicencia; }

set

{

if (!EdadParaLicencia(Edad, value))

{

throw new Exception($"La edad no es adecuada para el tipo de licencia {value}");

}

tipoLicencia = value;

}

}

public Chofer(string nombre, int edad, string tipoLicencia)

{

if (!EdadParaLicencia(edad, tipoLicencia)){

throw new Exception("La edad no es adecuada para el tipo de licencia {0}");

}

Nombre = nombre;

Edad = edad;

TipoLicencia = tipoLicencia;

}

private bool EdadParaLicencia(int edad, string tipoLicencia)

{

if ((tipoLicencia == "A" || tipoLicencia == "a") && edad >= 25)

{

return true;

}

if ((tipoLicencia == "B" || tipoLicencia == "b") && edad >= 23)

{

return true;

}

if ((tipoLicencia == "C" || tipoLicencia == "c") && edad >= 18)

{

return true;

}

if ((tipoLicencia == "M" || tipoLicencia == "m") && edad >= 16)

{

return true;

}

return false;

}

public void mostrarInformacion()

{

Console.WriteLine("El piloto es: {0}", Nombre);

Console.WriteLine("Edad: {0}", Edad);

Console.WriteLine("Tipo de Licencia: {0}", TipoLicencia);

}

}

}