Tarea: Herencia y Polimorfismo en C#

Continuación de lo realizado en el laboratorio que creamos para aprender a usar

Sobrescribir métodos en las clases derivadas

- Punto 1: Sobrescribir clase base Vehículo.
  En la clase base Vehículo añadí el método Frenar el cual le permite al Vehículo (la clase) poder reducir su velocidad. Lo que hice fue crear un método o propiedad marcado como virtual para que las clases hijo (por medio de herencia) la puedan usar, con un parámetro de tipo “int” llamado cuanto que representa la cantidad y dentro del método se resta valor a cuanto a la variable velocidad, así imprime un mensaje en la consola que muestra la nueva velocidad del vehículo en kilómetros por hora.

- Punto 2: Método adicional 1 a Vehículo.
  En la clase base Vehículo añadí el método Apagar el cual le permite al Vehículo poder apagar este o reducir su velocidad a cero. Lo que hice fue crear un método o propiedad marcado como virtual para que las clases hijo (por medio de herencia) la puedan usar, dentro del método se da el valor a velocidad a cero, así imprime un mensaje en la consola que dice que “El vehículo está apagado”.

- Punto 3: Método adicional 2 a Vehículo
  En la clase base Vehículo añadí el método Encender el cual le permite al Vehículo poder Encender este o iniciar este con in valor de cero. Lo que hice fue crear un método o propiedad marcado como virtual para que las clases hijo (por medio de herencia) la puedan usar, dentro del método se da el valor a velocidad a cero, así imprime un mensaje en la consola que dice que “El vehículo está encendido”.

Creación de 3 clases derivadas de la clase Vehículo

Auto de Combustión.

- Punto 4: Derivé la clase AutoDeCombustión de la clase Vehículo.

- Punto 5: Encapsulé la propiedad `private int cargaGasolina;` en la clase AutoDeCombustión.

- Punto 6: Encapsulé la propiedad `private int kilometraje;` en la clase AutoDeCombustión

- Punto 7: Encapsulé la propiedad `private int capacidadGasolina;` en la clase AutoDeCombustión

- Punto 8: Herede de la clase Vehículo a la clase AutoDeCombustión el método de acelerar, con un override añadí que cuando esta se use o que al usar el método esta adicional le reste a cargaGasolina.

- Punto 9: Herede de la clase Vehículo a la clase AutoDeCombustión el método de frenar, con un override añadí que cuando esta se use o que al usar el método esta adicional le reste a cargaGasolina.

- Punto 10: Basándome de la clase CarroElectrico creé el método `public int NivelGasolina() {return cargaGasolina;}` y asi poder crear el método `private void cargarGasolina;` para asi aumentar el nivel de gasolina.

- Punto 11: Cree el método de nivelKilometraje que mediante un public void me devuelve el nivel de kilometraje o recorrido.

- Punto 12: Cree el método de CapacidadGasolina que mediante un get permite leer el valor de la propiedad CapacidadGasolina. Cuando se accede a esta propiedad, se devuelve el valor de la variable privada capacidadGasolina.

Motocicleta

- Punto 13: Derivé la clase Motocicleta de la clase Vehículo.

- Punto 14: Encapsulé la propiedad `private int nivelGasolina;` en la clase Motocicleta.

- Punto 15: Encapsulé la propiedad `private int kilometraje;` en la clase Motocicleta

- Punto 16: Encapsulé la propiedad `private int encendida;` en la clase Motocicleta

- Punto 17: Herede de la clase Vehículo a la clase Motocicleta el método de acelerar, con un override añadí que cuando esta se use o que al usar el método adicional acelere más o aumente aún más de velocidad, y que le reste a cargaGasolina.

- Punto 18: Herede de la clase Vehículo a la clase Motocicleta el método de frenar, con un override añadí que cuando esta se use o que al usar el método esta adicional le reste a cargaGasolina, además que disminuye más de velocidad.

- Punto 19: En el método NivelGasolina, has definido una propiedad que controla el nivel de gasolina de la motocicleta. Este método asegura que el nivel de gasolina siempre esté dentro de un rango razonable, evitando valores negativos o excesivamente altos.

- Punto 20: Cree el método de nivelKilometraje que mediante un public void me devuelve el nivel de kilometraje o recorrido.

- Punto 21: He creado una propiedad Encendida para verificar el estado de la motocicleta y dos métodos (Encender y Apagar) para cambiar ese estado. La propiedad Encendida es de solo lectura pública, pero se puede modificar internamente mediante los métodos Encender y Apagar.

Camión

- Punto 22: Derivé la clase Camión de la clase Vehículo.
- Punto 23: Encapsulé la propiedad `private int cargaGasolina;` en la clase Camión.
- Punto 24: Encapsulé la propiedad `private int cargaActual;` en la clase Camión
- Punto 25: Encapsulé la propiedad `private int capacidadCarga;` en la clase Camión
- Punto 26: Herede de la clase Vehículo a la clase Camión el método de acelerar, con un override añadí que cuando esta se use o que al usar el método adicional acelere y que le reste a cargaGasolina.
- Punto 27: Herede de la clase Vehículo a la clase Camión el método de frenar, con un override añadí que cuando esta se use o que al usar el método esta adicional le reste a cargaGasolina, además que disminuye velocidad.
- Punto 28: Al igual que en AutoDeCombustión, basándome en el carro electrico creé el método `public int NivelGasolina() {return cargaGasolina;}` y asi poder crear el método `private void cargarGasolina;` para asi aumentar el nivel de gasolina.
- Punto 29: Cree el método de cargaActual que permite acceder y modificar el valor de este campo, con un get permito obtener el valor actual de cargaActual, se evalúa en CargaActual y esta se devuelve a cargaActual.
- Punto 30: He creado una propiedad que permite obtener y establecer la capacidad de carga del camión. Actualmente, el setter de esta propiedad es privado, lo que significa que solo puede ser modificado dentro de la clase Camión.

Resumen: En general lo que hice fue añadirle 3 métodos más a la clase principal (Vehículo), el método de frenado, encendido y apagado para que las clases derivadas de estas las pueda usar, como las 3 clases que creé para seguir el ejemplo que fueron la clase CarroDeCombustión, Motocicleta y Camión, en este creando 3 encapsulados similares y específicos para cada una. En CarroDeCombustión creé los encapsulados de cargaGasolina, kilometraje y capacidadGasolina, estas con sus métodos para usarlas y adicional derivando a esta de la clase principal métodos como acelerar y frenar en todas las demás

clases derivadas. En motocicleta creé los encapsulados de nivelGasolina, kilometraje y un bool encendida que me ayuda a prender y apagar de otra forma la clase o vehículo. En camion creé los encapsulados de cargaGasolina, cargaActual y capacidadCarga.