UNIVERSIDAD AUTÓNOMA "TOMÁS FRÍAS" INGENIERÍA DE SISTEMAS				
ESTUDIANTE:			**C.I.:**	
DOCENTE:	Ing. Ditmar Castro Angulo	**MATERIA:**	SIS-211 G2	**PRÁCTICA N4 AUX**
AUXILIAR:	Univ. Gabriel Alejandro Garvizu Salas	**FECHA:**	11/10/2024	

Ejercicio 1: Clases Abstractas e Interfaces en un Juego de Cartas

1. Crea una clase abstracta llamada "Carta" con los siguientes atributos y métodos:
   - Atributos:
     - valor (int)
     - palo (String)
   - Métodos:
     - Constructor para inicializar los atributos.
     - Método abstracto **jugar()**: este método será implementado por las clases derivadas.
     - Método **mostrarCarta()**: imprime el valor y el palo de la carta.

2. Crea una interfaz llamada "Accionable" con un método **realizarAccion()**. Esta interfaz puede representar una acción especial que algunas cartas pueden realizar.

3. Crea dos clases que extiendan "Carta":
   - "CartaNormal": Implementa el método **jugar()** para mostrar un mensaje que indique que la carta normal ha sido jugada.
   - "CartaEspecial": Implementa el método **jugar()** y también implementa la interfaz "Accionable". El método **realizarAccion()** debe imprimir un mensaje indicando la acción especial de la carta.

Ejercicio 2: Clases Abstractas e Interfaces en un Juego de Carreras

1. Crea una clase abstracta llamada "Vehiculo" con los siguientes atributos y métodos:
   - Atributos:
     - marca (String)
     - modelo (String)
     - velocidadMaxima (int)
   - Métodos:
     - Constructor para inicializar los atributos.
     - Método abstracto **acelerar()**: este método será implementado por las clases derivadas.
     - Método **mostrarInfo()**: imprime la marca, modelo y velocidad máxima del vehículo.

2. Crea una interfaz llamada "Turbo" con un método **activarTurbo()**. Esta interfaz puede ser implementada por vehículos que tienen la capacidad de usar turbo.

3. Crea dos clases que extiendan "Vehiculo":
   - "Coche": Implementa el método **acelerar()** para mostrar un mensaje indicando que el coche está acelerando.
   - "Moto": Implementa el método **acelerar()** y también implementa la interfaz Turbo. El método **activarTurbo()** debe imprimir un mensaje indicando que la moto ha activado el turbo.

Ejercicio 3: Clases Abstractas e Interfaces en un Juego de Simulación de Granjas

1. Crea una clase abstracta llamada “Animal” con los siguientes atributos y métodos:
   - Atributos:
     - nombre (String)
     - edad (int)
   - Métodos:
     - Constructor para inicializar los atributos.
     - Método abstracto **hacerSonido()**: este método será implementado por las clases derivadas.
     - Método **mostrarInfo()**: imprime el nombre y la edad del animal.

2. Crea una interfaz llamada “Productor” con un método **producir()**. Esta interfaz puede ser implementada por animales que producen algún tipo de producto (leche, lana, etc.).

3. Crea dos clases que extiendan “Animal”:
   - Vaca: Implementa el método **hacerSonido()** para mostrar un mensaje indicando que la vaca está haciendo su sonido característico. También implementa la interfaz “Productor”, donde el método **producir()** debe imprimir un mensaje indicando que la vaca está produciendo leche.
   - Oveja: Implementa el método **hacerSonido()** para mostrar un mensaje indicando que la oveja está haciendo su sonido característico. También implementa la interfaz “Productor”, donde el método **producir()** debe imprimir un mensaje indicando que la oveja está produciendo lana.

Ejercicio 4

Para cada uno de los tres ejercicios anteriores incorpore alguna característica propia, es decir, crear alguna interfaz extra, crear un nuevo método para hacer algo especifico, añadir atributos extra que vea conveniente, etc.
Cada ejercicio debe tener al menos una caracteriza extra funcionando y/o siendo usada

Ejercicio 5

Para cada uno de los tres primeros ejercicios incorpore una clase Main en la cual realice las pruebas de las clases creadas.

Ejercicio 6

Una vez completados los 5 ejercicios anteriores, es hora de realizar el ejercicio que vale más puntaje, hasta se podría decir que este ejercicio vale 100% de la práctica, puesto que si la carpeta con los distintos archivos no es subida a su repo de GitHub no podrá calificarse. Talvez ya te hayas percatado de que va, en caso de que no, ahí va.
Deberá de subirse los distintos archivos que se desarrollaron a lo largo de la práctica, a su repo en GitHub para su revisión, caso contrario no será válida la entrega por ningún otro medio.
Buena suerte y nos vemos en la siguiente práctica.

Fecha de entrega límite: 11/10/2024