Silvio Orozco Vizquerra
Carné 18282

Ejercicio 3

Identificación de errores:

Loos errores en el código de Saul, son los siguientes. Las propiedades dentro de la clase y que variables que sirven para instanciar el objeto avión deben de ser todas privadas. Además la variable dueño puede ser static pues el dueño es igual para todo el catálogo de aviones. Se debe agregar la cantidad de hélices que tiene el avión como otra variable. El método desplegar debería de ser método String y que de por retorno los datos necesitados para que estos puedan imprimirse en la pantalla principal. Le faltan 2 métodos de tipo constructor que le den parámetros al nuevo avión que se desea crear (Uno que contenga en parámetros también hélices y otro no.). Además debe tener el método de girar las hélices.
Ahora en código de la clase CatologoAviones, se necesita mejorar que el main puede interactuar con el usuario por medio de un scanner. Al usuario se le debe pedir que ingrese 3 distintos aviones y que indiqué todas las características necesarias del mismo. Al crear el avión se deben enviar dichas características a través de los parámetros y no como lo intenta instanciar Saul. Otra forma podría ser crear getters y setters. Finalmente, se debe poder desplegar la información de cada avión al igual que poder consultar un avión específico.

Análisis:

El programa debe mostrar la funcionalidad básica de ingreso y salida para el catálogo de coleccionista de aviones. Al sistema se debe poder ingresar nombre del fabricante, modelo, alcance máximo y si tiene hélices. Todos los aviones deben almacenar un campo con el nombre del dueño. Si tiene hélices, el avión permite girar sus hélices. Si lo desea se puede especificar un número de veces que gire, sino se realizará un número determinado de veces por defecto. El sistema debe de mostrar bajo demanda los datos de todos los aviones que tiene en memoria y debe poder consultar los datos de un avión especifico. Se debe presentar capacidad de almacenamiento para tres aviones distintos.
El código de Saúl falla en la expectativa de permitir al usuario ingresar los datos, no almacena campos de hélices ni incluye dichos métodos y no permite consultar los datos de un avión en específico.

Clases:

- Clase avión:
  - ✓ El objetivo de esta clase es crear un objeto de tipo avión que pueda instanciarse con parámetros iniciales para ejecutar ciertos métodos que muestren información del mismo en un catálogo.
  - ✓ Atributos:
    -Private piloto pilot; Nombre del piloto

-Private string fabricante; Nombre del fabricante.
-Private string modelo; El modelo del avión.
-Private double alcance; El alcance del avión.
-Static private String dueño; Quién es el dueño del catálogo y aviones.
-Private int helices; La cantidad de helices que tiene el avión.
-Private int vueltashelices; Cantidad de vueltas que puede dar una hélice.

- ✓ Métodos:
  -public String ToString(): Retorna un valor string que es un avión.
  -public String desplegar(): Retorna un valor string con las especificaciones del avión. Esto quiere decir que tendrá el fabricante, modelo, alcance, dueño, numero de helices y vueltas por hélice. Dependiendo si el avión tiene helices o no, desplegará dicha información.
  -public String girarhelices(): Da un valor de retorno si tiene helices y las vueltas que da, de lo contrario retorno un mensaje especificado que no tiene helices.

- Clase CatalogoAviones:
  - ✓ El objetivo de la clase es poder interactuar con el usuario y que se ingresen 3 tipos de aviones. De igual forma poder consultar un avión.
  - ✓ No tiene atributos.
  - ✓ Metodos:
    -public static void main(): Sirve para ejecutar nuestro driver program.
- Clase Piloto:
  - ✓ El objetivo de la clase es crear un tipo de persona que es un piloto para el avión.
  - ✓ Atributos
    Private string nombre;
  - ✓ Metodos:
    Private string pilotear(avión av); Muestra un mensaje que pilotea una nave si es el piloto de dicho avión.
- 3 clases de parte de avión que contenga motor, llave y alas. Que tengan únicamente el nombre.