Pablo René Arellano Estrada

151379

**Ejercicio No. 5:**

**Cambios:**

Los cambios realizados fueron respecto al ejercicio anterior fue la creación de una clase llamada vuelo, la cual tiene como atributos el fabricante, el modelo y el alcance, y no se había creado anteriormente porque no se utilizo el modelo view-controller-model.

Esta es llamada por una clase llamada GestionVuelo que asigna un objeto de tipo Avión a una ubicación de una lista para ingresar distintos parámetros. Anteriormente se ingresaban con distintas funciones con parámetros, lo cual eran más líneas de código, ahora solo son 5 lineas.

Asimismo se modifico la clase principal que no era GUI para que reciba los datos de los diez aviones y los muestre en un panel derecho por medio del uso de componentes y una clase Listener que luego permite que un método actionPerformed lleve a cabo distintas funciones.

**Análisis:**

|  |  |  |
| --- | --- | --- |
| **Clases:** | **Atributos:** | **Metodos:** |
| GUIaviones:  Es la clase encargada de recibir los datos usando una interfaz grafica formada por distintos componentes, la cual se comunica con la clase GestionAvion y MiBotonListener para realizar distintos eventos | * Frame: Ventana del usuario * tfFabricante: Cuadro de texto donde se ingresa el fabricante. * tfModelo: Cuadro de texto donde se ingresa el modelo. * tfAlcance: Cuadro de texto donde se ingresa el alcance. * btnAgregar: botón para agregar otro avion * btnCalcular: botón para llamar a un avión en específico * pResultado: Etiqueta donde se colocan los ingresos preliminares * lblCalculos: Etiqueta donde se coloca el avión escogido * lblError: Error por si no ingresa todos los datos en los cuadros de texto * gestionAvion: Es un objeto de tipo GestionAvion que tiene distintos métodos. * cont: cuenta las veces que se ingresa información a la lista * Cadena: cadena para imprimir los diferentes tipos de resultados separados por comas * MostrarUnico: variable entera ingresada en un cuadro de texto para deteminar que posición de la lista se buscara para mostrar los datos del avion | * Initialize(): es la que contiene todos los componentes que se usaran en al interfaz grafica. * Limpiar(): limpia los datos de los cuadros de texto para ingresar nuevos * Vacios(): Chequea que el cuadro de texto Fabricante tenga información y muestra una ventana nueva si no ha ingresado nada * noVacios():Chequea que todos los campos de texto tengan informacion * esNumero(): Chequea que los datos ingresados sean numeros |
| **Clases:** | **Atributos:** | **Metodos:** |
| GestionAvion: Clase encargada de crear la lista donde en cada ubicación se ingresara un objeto de tipo Avion. | * Avion: objeto de tipo avión que guarda el fabricante, modelo y alcance. * cantidadAsignada: es el limite que tendrá la lista * x: ubicación de la lista que se ingresa en un cuadro de texto. | * llenarAvion: es el método donde se crea una lista y que recibe como parámetros el fabricante, el modelo y el alcance, toda vez la lista no sea mayor a 10. * toString: es una cadena ordenada que llama a otra función que se encuentra en un un objeto de tipo Avion. |
| Avion: Clase que funciona de modelo de las características básicas de un avión. | * Fabricante: Empresa que produce avión * Modelo: tipo de avión * Alcance: Distancia máxima que alcanza un avión. | * GetFabricate, GetModelo y GetAlcance: pide los datos de los atributos * SetFabricante, SetModelo y SetAlcance: establece los datos ingresados de los atributos |
| MiBotonListener |  | * actionPerformed(): Realiza una acción cuando un JButton tiene valor true. |