

Comentarios del código

Clase Persona

En la clase persona, el atributo de asiento es por el tipo de encapsulación que se utilizo

Clase Avion

En la clase avión se utilizó una matriz dinámica de objetos tipo persona dinámicos, para mejor uso de memoria, y permanencia de datos. Tiene un constructor con parámetros donde el avión tendrá su tamaño en tiempo de ejecución, y así poder definir si el avión es grande o pequeño. No hay clase asientos ya que todo se puede hacer desde la clase avión, y así minimizar el tamaño del proyecto. Esta clase también tiene un constructor de copia `Avion(const Avion *a)`, este solo en caso de que se le asignen rutas con diferentes aviones a diferentes vuelos

Clase FlotaGestion

En esta clase se hizo un vector dinámico de objetos de tipo avión dinámicos, para permanecía de datos.

Clase Rutas

En esta clase se hizo un método de sumaRuta() en donde este suma las rutas de origen y destino y las devuelve en string para que no exista una relación entre la clase ruta y la clase vuelo.

Clase RutasGestion

En esta clase se hizo un vector dinámico con objetos de tipo rutas dinámicos, para permanecía de datos.

Clase Vuelos

En esta clase, se tiene un objeto de tipo Avion, porque los vuelos deben tener un avión asignado, y este previamente debe de estar registrado, por eso hay una dependencia entre estas dos clases, la clase flotaGestion y la clase vuelos. También tiene un método de muestraFecha(), que pide la fecha del avión y la guarda en un string.

Clase VuelosGestion

Esta clase tiene un vector dinámico con objetos tipo Vuelos dinámicos, para permanencia de datos. Este tiene un método revisaValor(char, int) donde este recibe por parámetros la fila y la columna, y revisa si los valores ingresados están en el tamaño del avión.

Clase Control

Esta clase incluye todos los .h de la solución, solo tiene métodos, y no tiene atributos. En esta clase hay 2 métodos, bienvenido() y hastaLuego(), donde se ve solo gráficamente el principio y final del programa.

Clase Interfaz

En esta clase solo muestra el menú y valida las opciones ingresadas. En la línea 115 del código en la columna 35, si se la cambia el valor de 4 por 5, puede ingresar métodos y evaluarlos desde el código.

MÉTODOS GENERALES

Los métodos `gotoxy()` y `col()` :

`Gotoxy()` :

Se puso por que a la hora de imprimir en la pantalla negra se imprime en forma de cascada y a partir de un `cin` o `getline`, deja de imprimir hasta ingresar un valor, es solo de manera grafica

`Col()` :

Se puso para cambiar los colores en la pantalla negra.

Habrán comentarios en el código explicando los métodos o variables.

Hay métodos de `Sleep(int)` así que si no dice “presione cualquier tecla para continuar...” no hay que tocar nada, por que después se corre el buffer por teclado.