

Laboratorio 3 Herencia

Síntesis

El programa realizara las siguientes opciones y trabajos:

- Crear una estructura de datos, de dos tipos de datos, siendo los Básico y VIP. Junto a la información normal de usuario.
- Sera capaz de leer archivos CSV con las características provistas y los nombres adecuados.
- También deberá ser capaz de crear un CSV y almacenar datos dentro de el
- Buscar Usuarios y guardar datos de ellos mismos
- Hacer movimientos o “registros” a nombre de un usuario
- Diferentes tipos de usuario indican diferentes formas de movimiento.

Clases y propiedades

- Main (Public)
El menú principal, con todos los ordenamientos estadísticos con excepción de los métodos de efectividad, tiene el lector de CSV al principio, y se ejecuta el escritor al salir del programa.
- Básico (Private)
Constructor tipo Private, el cual hereda de la clase Movimiento, esta estará encargada de provisionarle los métodos a todos los objetos que son Basicos, siempre incluyendo los datos de Movimiento, agregamos: **Cupón. Implementa Base.**
- VIP (Private)
Constructor tipo Private. Al igual que Básico, utiliza esos datos, pero en este caso no agregamos más atributos a sus objetos. **Implementa Base.**
- Usuario (Private)
Constructor tipo Private. Él es el encargado de guardar el nombre del usuario a la vez que su contraseña, de forma que podamos verificar a nombre de quien se está haciendo las reservas de avión. Tiene los atributos: **Nombre, contraseña, clase de plan (Básico o VIP).**
- Base (Interfaz)
Una interfaz que será implementada en Básico e VIP, ambos tendrán los siguientes métodos setters conjunto a sus getters. **serVueloInicio (), setVueloFinal (), setBoletos (), setAerolinea(), serTarjeta(), Modificar Contra(), set Cuotas(), setClaseVuelo(), AsignacionAsiento(), y setNumMaleta().**

Parámetros

- Tendremos diferentes instancias de otros objetos que heredan de Movimiento y a la vez que implementan Base, estos métodos harán algo entre las mismas líneas de las mismas cosas.

Explicación a detalle

Tendremos un arraylists de tipo interfaz para poder usar ambos tipos de objetos. Cuenta que estoy agregando el otro objeto por que se siente raro que no este, aunque no sea mi trabajo. En esencia lo que hasta el sistema será primero identificar que tipo de dato será el usuario, si será parte de VIP o de Básico. Cuando se identifica el tipo de usuario que estamos utilizando ya se harán las demás funciones debidamente. Los horarios los tendrá que escribir la persona, la vez de boletos y otras cosas. En general, todos los datos son escritos. Después de tomar en cuenta los datos, confirmaremos las cualidades del viaje, cosas como boletos, asiento y maletas. Una vez tengamos en cuenta que todo está en orden, ahí es donde lo guardamos como uno de dos tipos diferentes de movimientos.

URL del repositorio en GitHub: <https://github.com/anthonylouschwank/Laboratorio-4>

