

TP N°3

Bacchetta, Tomás

El programa "Cuartel" permite dar de alta, modificar, y eliminar soldados, que a su vez pueden asignarse a escuadrones. Para estos escuadrones luego, se simulan expediciones, de donde se obtendrán datos aleatorios que serán cargados en los atributos de los soldados y de los escuadrones.

Los escuadrones tienen un límite de 4 soldados, y todos deben ser de clase diferente. Las clases se eligen al momento de dar de alta un soldado, y determinan también las chances de conseguir diferentes atributos a la hora de realizar las expediciones.

Las estadísticas de los soldados se borran a medida que se los va expulsando, pero la de los escuadrones es persistente (no importa si se expulsa a todos sus integrantes).

El estado del programa puede guardarse en formato .xml o .json. El informe principal se puede guardar como texto plano .txt

Dentro de la carpeta de la solución hay un archivo .xml llamado TESTCARGA, donde ya hay datos cargados con los que se puede corregir más fácilmente.

Implementación de la teoría:

Clase 10: Excepciones

En los archivos de clases ArchivoIncorrectoOCorruptoExcepcion.cs y ErrorDeArchivoExcepcion.cs se definen excepciones propias, que son lanzadas en Archivo.cs y sus clases derivadas. Luego se usan tries en el código de FrmCuartel.cs, en los métodos de apertura y guardado de archivos.

La primera excepción está pensada para la apertura de archivos (puede ser que un archivo fuese editado a mano y arruinado y el programa podría, en primera instancia, tratar de leerlo, por ende es conveniente comprobar que sea compatible con el programa).

La segunda excepción se encarga del guardado, que el directorio exista, que sea válido, y que no esté vacío.

Es imprescindible controlar excepciones en los apartados de carga y guardado ya que la complejidad de la interacción con el sistema operativo y las unidades de almacenamiento son caldo de cultivo para diversos errores a veces no tan evidentes.

Clase 11: Pruebas Unitarias

Se prueban algunas sobrecargas de operadores de las clases Cuartel y Escuadrón. Se encuentran en TestMiscelaneos.cs, en proyecto Test.

Clase 12: Tipos Genéricos

Se utilizan tipos genéricos para el manejo de archivos, ya que parto del desconocimiento de qué tipo de objeto voy a guardar. En las clases derivadas de Archivos.cs son utilizados los tipos genéricos, y son asignados a un tipo a la hora de instanciar estas clases, tanto en los métodos

Abrir(), Guardar(), GuardarComo() de clase FrmCuartel.cs, como en el método btnGuardarInforme_click de la clase FrmInforme.cs, aunque en ese caso al ser txt, el tipo admitido va a limitarse a ser string. También la interface IManejoDeArchivos utiliza tipos genéricos ya que los métodos que define pueden devolver o recibir cualquier tipo (limitado a tipo de clase)

Clase 13: Interfaces

También se utilizan interfaces con IManejoDeArchivos, que naturalmente se encarga de definir los métodos que van a utilizar las clases derivadas de la clase abstracta Archivo. Por las razones expuestas más arriba, también se la combina con los tipos genéricos

Clase 14: Archivos y Serialización

En Archivo.cs y sus clases derivadas como así también la interfaz IManejoDeArchivos, se permite la interacción con archivos, tanto por serialización (para guardar y abrir el estado del programa) como por texto plano (para guardar el informe). Se utilizan lógicamente también en FrmCuartel.cs (para serialización), en FrmInforme.cs (y para guardado de archivos). Como no guarda información de clases polimórficas, se puede elegir tanto xml como json para el guardado y apertura del estado de todo el proyecto, aunque de ser diferente el caso se podría usar xml para un guardado general, y json para un guardado de algún dato específico.