Alumno: Luciano Agustin Crocco

División: 2C

Trabajo Practico 3 – Estadísticas para un sistema de escuderías y pilotos

Funcionalidad pretendida

El proyecto consta de un formulario principal donde tendrán dos botones para generar una escudería o un piloto. Trabajo con dos listas separadas para ambos casos.

La idea del sistema es dar datos de ingreso al usuario, la posibilidad de cargar y guardar listas, de poder generar estadísticas (máximo, mínimo, promedio) de pilotos o escudería y guardar dichas estadísticas en un archivo TXT.

Las estadísticas las divido en 3 partes

1. Estadísticas de todos los pilotos cargados.
2. Estadística de todos los pilotos cargados a una escudería.
3. Estadística de las escuderías cargadas.

Pequeño resumen de los temas aplicados TP3

- *Excepciones:* Hay una biblioteca aparte para las excepciones, las utilizo sobre todo en las sobrecargas de las clases y en la serialización de archivos.

- *Test Unitarios:* Generé un proyecto de prueba de MSTest con 8 test unitarios distintos donde probé funcionalidades como ExpectedException y AAA (Arrange, Act, Assert).

- *Genéricos:* Utilice el concepto de tipos genéricos tanto para la serialización de datos XML y JSON (visto en clase) como para el formulario FrmEscuderia. Como no voy a utilizar otra clase de Escudería decidí, por el momento, desechar el concepto de una lista genérica. Esto se debe mas que nada a un error mío a la hora de encarar el proyecto y cambiarlo me puede llevar muchísimas horas. Por falta de tiempo decidí no realizar los cambios.

- *Archivos:* Utilice archivos para guardar estadísticas o para guardar archivos JSON.

- *Serialización:* Serialice la lista de pilotos con XML y para la lista de escuderías utilicé el tipo JSON ya que XML tenía distintos problemas a la hora de serializar una clase heredada (Busque una solución, pero la verdad me iba a consumir muchísimo tiempo. En SO alguien tuvo un problema similar y su corrección (https://stackoverflow.com/questions/12237268/how-to-implement-xml-serialization-with-inherited-classes-in-c-sharp)

- *Interfaces:* Aplique interfaces de tipo genéricas para Archivos y Serialización lo que me permitió tener funcionalidades agrupadas e implementaciones diferentes según la necesidad.

Trabajo Practico 4 – Continuación

Continuación

Continuo con la misma idea explicada anteriormente. Añado y edito funciones para aplicar los temas vistos que pide el TP4. Edito mi aplicación de Generics en formularios, explicado en la sección de TP3.

Realice el cambio pedido en la anotación del TP3 (Hacer que cuente con información precargada (podrían ser los archivos para importar), es muy largo poder realizar una prueba cada vez que se ejecuta) en el evento load del formulario principal.

Pequeño resumen de los temas aplicados TP4

- *Base de Datos SQL:* Apliqué SQL para:
  - Guardar todos los pilotos que genere la aplicación desde que se lanza
  - Editar pilotos cargados en memoria, matcheando con el SQL para editarlos también en la base de datos.
  - Cargar pilotos del historial a la lista de pilotos actual.
  - Eliminar un piloto del historial.
  - Estadísticas del historial de pilotos.

  Base de datos adjunta, para conectarse cambiar el Data Source en: Entidades->Base De Datos -> PilotoBDD.cs -> Constructor

- *Delegados:* Utilicé delegados para lanzar Hilos de ejecución como, por ejemplo, recargar elementos de un formulario en el hilo principal desde un hilo secundario. También para encapsular manejadores de Eventos.

- *Hilos:* Reemplace las funciones tanto de instancia como estáticas que refrescaban la list box de mis formularios. Estas funciones fueron reemplazadas por hilos. También utilicé hilos para lanzar Eventos, entre otras cosas.

- *Eventos:* Hice un evento que cuenta la cantidad de pilotos dentro de la lista y refresca un lbl constantemente.

- *Métodos de extensión:* Realmente no necesité uno, aún así generé un contador de caracteres y añadí un máximo, mínimo, y promedio de caracteres por nombre y apellido a las estadísticas.