



Ejercicios

Arrays y colecciones

Ejercicios de arrays y colecciones

Antes de empezar

- Deberá asignarle a la propiedad 'Title' de la clase Console, el número de ejercicio.

Por ejemplo:

Console.Title = "Ejercicio Nro ##" donde ## será el número del ejercicio.

- Del mismo modo se deberán nombrar los proyectos de consola.

Por ejemplo:

Ejercicio_##.

- Para visualizar los valores decimales de los ejercicios, deberá dar el siguiente formato de salida al método Write/WriteLine: "#,###.00".

I01. Números locos

Crear una aplicación de consola que cargue 20 números enteros (positivos y negativos) distintos de cero de forma aleatoria utilizando la clase `Random`.

1. Mostrar el vector tal como fue ingresado.
2. Luego mostrar los positivos ordenados en forma decreciente.
3. Por último, mostrar los negativos ordenados en forma creciente.

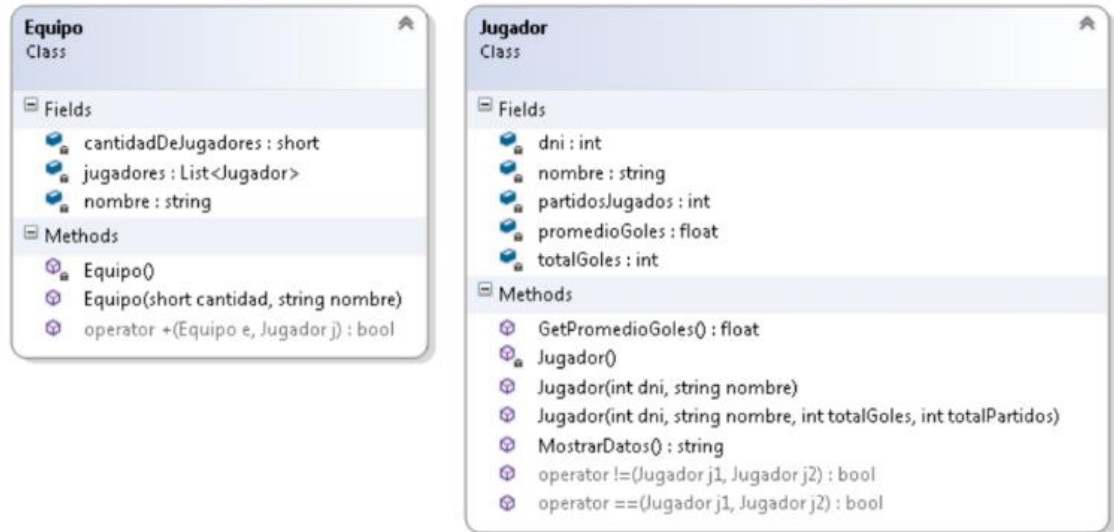
I02. Números locos II

Crear una aplicación de consola que cargue en una Lista, Pila y Cola 20 números enteros (positivos y negativos) distintos de cero de forma aleatoria utilizando la clase `Random`.

1. Mostrar la colección tal como fue cargada.
2. Luego mostrar los positivos ordenados en forma decreciente.
3. Por último, mostrar los negativos ordenados en forma creciente.

I03. Estadística deportiva

Crear un proyecto de biblioteca de clases que contenga las clases `Jugador` y `Equipo`:



Clase Jugador:

- Todos los datos estadísticos del jugador se inicializan en 0 dentro del constructor privado.
- El promedio de gol sólo se calculará cuando invoquen al método `GetPromedioGoles`.
- `MostrarDatos` retornará una cadena de string con todos los datos y estadísticas del jugador.
- Dos jugadores serán iguales si tienen el mismo DNI.

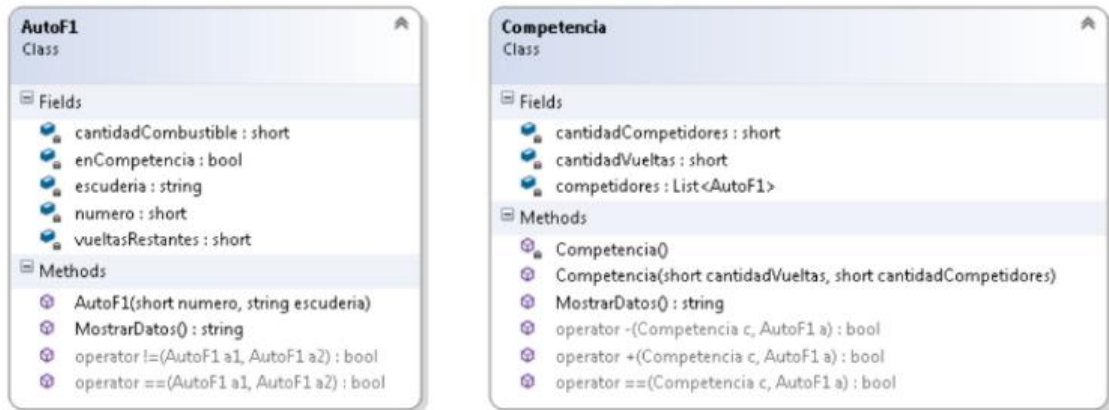
Clase Equipo:

- La lista de jugadores se inicializará sólo en el constructor privado de `Equipo`.
- La sobrecarga del operador `+` agregará jugadores a la lista siempre y cuando no exista aún en el equipo y la cantidad de jugadores no supere el límite establecido por el atributo `cantidadDeJugadores`.

Crear un proyecto de consola y generar las invocaciones necesarias en el método `Main` para probar el código.

I04. Enciendan sus motores

Crear un proyecto de biblioteca de clases que contenga las clases `Competencia` y `AutoF1`:



Clase AutoF1:

- Al generar un auto se cargará el atributo `enCompetencia` como falso. Inicializar `cantidadCombustible` y `vuelasRestantes` en 0.
- Dos autos serán iguales si los atributos `número` y `escuderia` son iguales.
- Realizar los métodos getters y setters para `cantidadCombustible`, `enCompetencia` y `vuelasRestantes`.

Clase Competencia:

- El constructor privado será el único capaz de inicializar la lista de competidores.
- La sobrecarga del operador `+` agregará un competidor si es que aún hay espacio (validar con `cantidadCompetidores`) y el competidor no forma parte de la lista (`==` entre `Competencia` y `AutoF1`).
- Al ser agregado, el competidor cambiará su estado `enCompetencia` a verdadero, la cantidad de `vuelasRestantes` será igual a la `cantidadVueltas` de `Competencia` y se le asignará un número aleatorio entre 15 y 100 a `cantidadCombustible`.

Crear un proyecto de consola y generar las invocaciones necesarias en el método `Main` para probar el código.