# Examen2ª Evaluación

## **(DOSSIER - 1,5 pt)** Encontrar cinco errores de normas de estilo en el fichero *loto.cs,*indicando número de línea, error encontrado y solución.

* Nombre de clase debe utilizar estilo PasCal: línea 7. Error loto. Sol LotoDAS2223.
* Nombre de constantes deben utilizar estilo PasCal: líneas 10 a 12. Errores MAX\_NUMEROS, NUMERO\_MENOR, NUMERO\_MAYOR. Sol MaximoNumeros, NumeroMenor, NumeroMayor.
* Nombre de campos debe utilizar estilo caMel: línea 14. Error int[] \_nums. Sol private in[] numerosCombinacion. Ahora además podemos eliminar el comentario que explica qué es ese campo.

Línea 15. Error ok. Sol combinacionValida.

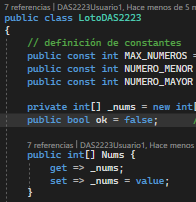
**Otros errores:**

* Error comentario no existente: línea 23. Error “//(sin comentario)”. Sol Eliminar “//”.
* Espaciado: línea 27 (entre otras muchas –línea 32, 40,…-): Error i=0. Sol i = 0;

## **(DOSSIER+COMMIT - 1,5 pt)** Si existen, detectar y aplicar al menos tres patrones de refactorización (tanto en el fichero *Loto.cs* como en el fichero *Form1.cs*), indicando el patrón que se aplica y, si es posible aplicarlo con Visual Studio, la opción que se usa.

**Clase Loto**

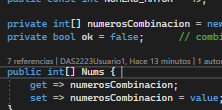
Vemos que uno de los campos es público. Es buena idea encapsular los campos para poder restringir su acceso desde fuera. Se validarán el método de su propiedad, que recogerá y dará valor a estos con sus métodos get y set.



En VS, si hacemos clic en esa líne, y vamos a Editar/Refactorizar/Encapsular Campos… nos lo hace automáticamente.

Renombrar también es una técnica de refactorización, y aquí vamos a utilizarla en la línea 14 para cambiar el nombre de \_nums a numerosCombinacion (campos en caMel).

Vamos a vamos a Editar/Refactorizar/Cambiar nombre y hacemos lo dicho. Resultado:



**Form1**

Encapsularemos los campos miLoto y miGanadora:

