Universidad Tecnológica Nacional Facultad Regional Avellaneda										
Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos										
Materia: LABORATORIO II										
Apellido:						Fecha:				
Nombre:						Docente(2):	NEINER			
División:						Nota(2):				
Legajo:						Firma(2):				
Instancia(1):	PP		RPP		SP		RSP		FIN	X

1) **(1 pto)** Realizar un método de extensión, llamado ***EsPar***, para la clase Int32, que permita determinar si el número es par o no.
Crear otro método de extensión, llamado ***EsImpar***, para la clase Int32, que determine si el número es impar o no. Reutilizar código.

2) **(1 pto)** Crear un objeto de tipo Stack<Double>. Apilarle la siguiente secuencia de números: 1, 2, 3. Realizar un algoritmo que permita tener la secuencia ordenada de manera inversa en la misma colección, es decir: 3, 2, 1. De ser necesario, utilizar sólo colecciones de tipo Stack<Double> ó Queue<Double>.

3) **(2 ptos)** Crear dos objetos de tipo Deposito, cada uno de estos objetos contiene una lista de la clase Producto. La clase Producto tiene dos atributos: Nombre y Stock.
Se debe poder sumar las listas de los dos depósitos (con la sobrecarga de un operador en la clase Deposito) y guardar el valor que retorna en una lista de Productos, recordar que si un producto está en las dos listas, se debe sumar el stock y no agregar dos veces al mismo producto.

4) **(4 ptos)** Crear la clase Galpon, que contenga una lista genérica de tipo T, con una propiedad ***"Cantidad"*** que sólo permita asignar un valor (entero) al atributo ***"_cantidad"*** y un evento (diseñarlo para que reciba un Object y un EventArgs, su retorno será void). Si el valor que se intenta asignar es cero, se deberá lanzar una excepción de tipo ArgumentException informando de lo acontecido. Si el valor es par (utilizar lo hecho en el punto 1), dejar asignarlo. Si el valor es impar (utilizar lo hecho en el punto 1) disparar el evento ***EsImpar*** cuyo manejador tendrá que escribir en un archivo de texto (log.txt) la fecha (hh:mm:ss y el valor) y asignarlo.

5) **(1 pto)** Realizar una estructura try-catch (en el Main) para la escritura de la propiedad del punto anterior que, al capturar la excepción, muestre el mensaje.

6) **(2 ptos)** Realizar el burbujeo de una excepción **propia**, comenzando en un método de instancia, pasando por un método de estático y capturado por última vez en el Main.

7) **(3 ptos)** Crear la siguiente interface: `public interface IGuardarXML { bool SerializarXML(); }`
Implementarla en la clase Galpon.
Agregar a una instancia de tipo Galpon<Deposito> (que contenga al menos un objeto de tipo Producto, otro de tipo ProdImpuesto, otro de tipo ProdExport y otro de tipo ProdVendido) y generar una serializacion XML del galpón. Modificando lo que crea conveniente para poder serializar todos los atributos de todos los objetos intervinientes, guardando en el archivo ***archivo.xml***.

TIEMPO MAXIMO PARA RESOLVER EL EXAMEN 60 MINUTOS.