**Ejercitación 2 Formularios – C#**

1. Datos (estructura) de la persona (clase): nombre y apellido, teléfono, edad, dirección. Utilizar constructores de estructura donde debe presentarse sobrecarga para datos con y sin teléfono. Realizarlo para 5 personas.
2. Utilizar 3 clases: Triángulo, Rectángulo, Circunferencia. Averiguar a través de métodos: perímetro y superficie de cada figura.

Crear la función Desplegar() donde se presente un menú con las siguientes opciones: 1- Triángulo, 2- Rectángulo, 3- Circunferencia.

Crear la función Calcular() donde se presente un menú con las siguientes opciones: 1- Perímetro, 2- Superficie.

1. Idem2, realizar el ingreso de datos por teclado mediante la función IngresarDatos().
2. Idem2, determinar mediante el método EsCuadrado() si el rectángulo es un cuadrado o no.
3. Idem2, realizar el método EsTriangulo() teniendo en cuenta que la suma de los dos lados menores tiene que ser superior al lado mayor para que formen un triángulo, indicar si los números indicados forman un triángulo. Si lo forman, mediante el método TipoTriangulo() determinar de qué tipo es (según sus lados).
4. Idem2, para el caso de un triángulo equilátero, calcular su perímetro solo ingresando el valor de un lado y calcular su superficie con el método CalcularAlt().

La altura de un triángulo equilátero es: lado \* (√3) / 2

1. Gestión de un supermercado.

Se desea automatizar el proceso en un supermercado. El mismo cuenta con 40 cajas registradoras y cada una de ellas registra durante la jornada toda la información de las ventas efectuadas desde las cajas; por cada venta (artículo que pasa por caja) se registra la información:

número de caja, código de artículo, cantidad de unidades vendidas

Al final del día se utiliza la información registrada en las 40 cajas para actualizar las existencias globales del supermercado en función de los artículos vendidos. El inventario de existencias guarda por cada artículo a la venta la siguiente información:

código de artículo, stock, precio por unidad

Se pide realizar al final del día:

* + Venta total por caja (dinero)
  + Venta total de artículos (código y cantidad por caja)
  + Stock de artículos.

1. Estacionamiento.

Para hacer frente a los problemas de estacionamiento en la ciudad, la municipalidad está construyendo un moderno parking automático. El estacionamiento cuenta con 3 pisos (PB, 1er. piso y 2do. piso).

Cada piso se identifica por su número y capacidad máxima. Cada estacionamiento se identifica por su número de piso y puede tener dos estados (ocupado o libre).

Una aplicación informática controlará el estacionamiento de los vehículos. Un coche se identifica por su patente.

Cuando un usuario llega con intención de estacionar su coche, el sistema comprueba si queda alguna plaza libre, de ser así busca una (considerando planta baja como 1era opción, 1er. piso, 2do, piso).

Al momento que un conductor retire su vehículo, debe liberarse ese estacionamiento, quedando vacante para utilizar.

Realizar la codificación que realice lo pedido.

1. Gestión de la biblioteca

Se quiere construir una aplicación para gestionar una biblioteca. Dicha aplicación debe guardar la siguiente información:

* + 1. Relación de todos los socios. Para cada socio se tiene su nombre, el número de socio, dirección, y la referencia de los ejemplares que tiene en préstamo. Se pueden tener 2 libros en préstamo como máximo.
    2. Fondos de la biblioteca, donde se recopila la información de todos los ejemplares de la biblioteca. Por cada libro se guarda su código de identificación, título, autor, estado (libre, prestado o desaparecido).
    3. Relación de los ejemplares prestados. Por cada ejemplar prestado se guarda su código, el número de socio que lo tiene en préstamo, la fecha del préstamo y la fecha de devolución.
    4. Relación de morosos, es decir aquellos socios que no han devuelto alguno de los libros que tienen en préstamo (y que debieran haber devuelto)

Realizar:

* + - * Las operaciones básicas que debe permitir realizar la aplicación son: registrar un nuevo socio, borrar un socio, prestar un libro y devolver un libro, además de las correspondientes a la gestión de las relaciones ya mencionadas de ejemplares prestados y de morosos.
      * Listado de libros prestados y los socios que tienen los mismos.
      * Dada una cierta fecha, listar los libros que deben entregarse hasta ese día inclusive.
      * Implementar la operación PrestarLibro. Ésta es una de las opciones del menú principal de la aplicación. Implementar esta operación implica comprobar que el libro se puede prestar, y que el socio no es moroso ni tiene en préstamo el número máximo de libros permitido.

1. Venta de vehículos. Clase abstracta.
2. Colegio. Interfaces, propiedades.
3. Proveedor de internet. Paso de parámetros al menos de 2 clases.

**Listas**

1. Realizar un programa ingresando números enteros mientras no se introduzca el cero o hasta 20 números y rellenar dos listas, uno con los números pares y otro con los números impares. Al final se debe mostrar por pantalla ambas listas.
2. Realizar un programa que lea los tiempos en los que de 10 corredores han acabado una carrera. El programa debe determinar cuáles son corredores del primer, tercer y quinto puesto, así como cuál es el tiempo medio en que se ha corrido la carrera.
3. Ingresar 5 números decimales, ordenarlos de menor a mayor.
4. Ingresar 4 nombres y ordenarlos alfabéticamente.
5. Se ingresan 6 (seis) números, se deben ir armando las listas nodo a nodo, insertando cada número en su nodo correspondiente, imprimir lista de menor a mayor.
6. Se ingresan 6 (seis) números, si el número es par se elimina el último nodo, si el número es impar se guarda en la lista. Imprimir lista resultante. Validar casos donde no hay nodo para eliminar.
7. Ingreso de números, cada 3 números que se insertan se quita uno de la lista, imprimir la lista resultante (pila).
8. Ingreso de números, cada 3 números que se insertan se quita uno de la lista, imprimir la lista resultante (cola).