

Programación Orientada a Objetos.

Relación Nº: 7

1. Escribe una Librería (DLL) en la que se incluyan la clase MenuPrincipal que tenga varios métodos sobrecargados que permitan mostrar un menú en las posiciones de coordenadas indicadas, pasándole también un array de string con las opciones para mostrar el menú. Debe tener otra clase Marcos que incluya varios métodos que te permitan dibujar en pantalla un marco de trazo simple o doble de algunos colores y en las posiciones indicadas por sus vértices 1 (sup-izq) y su vértice 3 (inf-dcho). Debes de controlar que si las dimensiones pasadas son mayores que las que soporta la pantalla producirá un error que debemos solucionar, propongo la opción siguiente: si las dimensiones de alto o ancho del marco está fuera de la pantalla dibujaremos un marco de tamaño máximo disponible, según sea para alto o ancho.
2. Escribe un programa que use y pruebe las clases y métodos del ejercicio anterior, escribe cada clase en un fichero y ponle ayuda XML a la clase y a los métodos. También puedes sobrecargar o codificar algunos métodos del ejercicio anterior para que muestre el menú principal en distintos colores de colores para las letras, fondos, etc. Usa la imaginación.
3. Escribe una aplicación con estos dos espacios de nombres: Geometría y PruebaGeometria. Dentro del espacio de nombres Geometría tienes que escribir dos clases: Punto y Cuadrado. La clase Punto ha de tener dos campos de sólo lectura: X e Y (que serán las coordenadas del punto). La clase Cuadrado ha de tener las siguientes propiedades del tipo Punto (de solo lectura): Vertice1, Vertice2, Vertice3 y Vertice4 (que corresponden a los cuatro vértices del cuadrado). La base de todos los cuadrados de esta clase será siempre horizontal. También ha de tener las propiedades Lado, Área y Perímetro, siendo la primera de lectura/escritura y las otras dos de sólo lectura. Por otro lado, debe tener dos constructores: uno para construir el cuadrado por medio de los vértices 1 y 3 y otro para construir el cuadrado a través del Vertice1 y la longitud del lado. En el espacio de nombres PruebaGeometria es donde escribirás una clase con un método Main para probar si funcionan las clases escritas anteriormente. En este espacio de nombres quiero que utilices la directiva using para poder utilizar todos los miembros del espacio de nombres Geometría directamente. En este espacio de nombres escribe también un método que muestre todos los datos de un cuadrado en la consola. Podrías usar los métodos del ejercicio 1 para, además de mostrar los datos del marco, dibujarlo en pantalla.
4. Haz un programa que use una clase Números que tenga los métodos: FactorialRecursivo, FactorialIterativo, FibonacciIterativo, FibonacciRecursivo, SumaNumerosIterativos, SumaNumerosRecursivo y EsPrimo. Haz un Menú de Consola o Interfaz de gráfico para Windows, para probar los métodos de la clase creando objetos de ella. Los métodos propuestos son que tienes hecho en la relación de ejercicio número 5. Para hacer el menú usa lo que has hecho en los ejercicios anteriores.
5. Haz un programa, usando clases, para gestionar una colección de estructura ordenada por código y sin repetición, los campos mostrados son...

Código	Entero, único sin repetición
Apellidos	Cadena de texto
Nombre	Cadena de Texto
Fecha Nacimiento	Tipo Fecha
Sueldo real	Real

El programa debe de tener como mínimo las opciones: *Añadir, Buscar, Ver, Listar y Borrar*.

6. Haz un programa que tenga una clase CFechas que gestione fechas correctas incluyendo los años bisiestos. Dado el día año y mes se comprueba que son correctos se decide si es bisiesto el año y se devuelve la fecha corta con formato dd/mm/aaaa o con formato largo “*día de mes del año*”.
7. Haz un programa que tenga, una Interfaz (ICarrera) con un método Correr() y dos clase (Coche, Deportista) que implemente el interfaz indicado. El método Correr() de la primera clase mostrará los datos del coche y la frase “El coche está en marcha” y el de la clase Deportista además de los datos del deportista mostrará la frase “El corredor de fondo está en ello...”. Usa el polimorfismo para llamar al método Correr.

8. Queremos hacer una aplicación para gestionar un conjunto de vehículos (motos, coches y bicicletas). Las bicicletas pueden ser de dos tipos de montaña o de paseo. Todos los vehículos poseen (nombre, nºruedas, color, tipo_tracción), las motos tienen (potencia, tipo_combustible: mezcla, normal), los coches tienen (velocidad máxima, estado: marchando o parado) y la bicicletas tienen en común el precio y fecha de compra, las de paseo además tienen nº de cestas, modelo y marca. Las de montaña tienen tipo_de_amortiguación, kit de reparación y diámetro de las ruedas.

Haz las clases (aplicando los conceptos estudiados de herencia, etc), para implementar este ejercicio con los datos solicitados. Haz una demo que compruebe, usando la POO, que todo funciona según lo previsto.

Este diagrama te da una idea de lo que hay que implementar.

