EJERCICIOS MÓDULO 9

Ejercicio 1

a) Crea una clase Alumno que contenga los atributos:

- nombre
- apellidos
- edad

y los siguientes métodos:

- Constructor por defecto
- Constructor con 3 parámetros, que inicialicen los atributos de la clase
- Get y set necesarios para consultar y modificar los atributos
- ToString, que permita mostrar estos datos por pantalla.

b) Realiza un programa que cree un objeto alumno llamado alumno1 mediante el constructor por defecto y posteriormente vaya pidiendo al usuario que introduzca los diferentes valores de sus atributos. Debe usar los métodos set para asignar estos valores.

c) Añade un objeto alumno llamado alumno2 mediante el constructor con 3 parámetros. Anteriormente debe haber pedido al usuario que introduzca los diferentes valores de sus atributos, que utilizará en el constructor.

Ejercicio 2

a) Realiza ahora los métodos de acceso a los atributos de la clase Alumno mediante los descriptores de acceso get y set, que se consideran propiedades de lectura y escritura.

b) Realiza un programa que cree un objeto alumno mediante el constructor por defecto y utilice los nuevos descriptores de acceso para asignar estos valores desde teclado. Al finalizar deberá imprimir las características del alumno, mediante el método ToString():

Ejercicio 3

Extiende el ejercicio 1 añadiendo una propiedad nota_examen que contenga un valor numérico. Define el método que permita obtenerlo así como el método que permita establecerlo. En este caso se debe controlar que la nota sea correcta, es decir, entre 0 y 10. Por ejemplo si es mayor que 10, debe ser 10. Si es menor que 0 debe ser 0.

Ejercicio 4

Extiende de nuevo el ejercicio 1, creando una propiedad llamada sex que podrá contener “H” o “M” según sea el alumno hombre o mujer. Añade métodos que permitan acceder y poner su valor.

Ejercicio 5

a) Amplía la clase alumno, de manera que ahora contenga una propiedad en forma de array que coleccione 10 instancias de la siguiente clase llamada Asignatura:

```
public class Asignatura {
    private string Nombre;
    private string Area;
    public Asignatura(string sNom, string sAmbito) {
        this.Nombre = sNom;
        this.Area = sAmbito;
    }
    public void setNombre( string sNom ) { Nombre =sNom;}
    public string getNombre() { return Nombre; }
    public void setAre(string sArea) { Area = sArea; }
    public string getArea() { return Area; }
}
```

b) Crea un método MuestraAsignaturas que muestre por pantalla las diferentes asignaturas a las que está apuntado el alumno.

Ejercicio 6

Crea una nueva clase Aula que permita contener instancias de la clase Alumno (por ejemplo mediante un atributo en forma de array). Debe definir los siguientes métodos:

a) Método addStudent que registrará un alumno en la clase Aula.

b) Método calcularMedia debe permitir calcular la media de los alumnos del aula, a partir de su nota de examen del ejercicio 2, y mostrarla por pantalla.

c) Métodos getNumHombres y getNumMujeres que retornan por pantalla cuántos hombres y cuantas mujeres hay en el aula, respectivamente.

d) Realiza un programa que cree 3 instancias de alumno, las registre en la aula creada y posteriormente calcule la media de los alumnos y muestre el número de hombres y mujeres existentes en el aula.

Ejercicio 7

a) Crea una clase Academia que contenga instancias de la clase Aula. La clase Academia tendrá 3 métodos:

- Método addAula() para añadirle instancias de aulas.
- Método getMediaAulas() para obtener un informe de la media de las aulas.
- Método resumenSexoEstudiantes() para averiguar cuántos alumnos y alumnas hay en la academia.

b) Realiza un programa que defina 6 alumnos, los reparta entre 2 clases. A continuación creamos academia e imprimiremos el resultado de la llamada a los métodos getMediaAulas() y resumenSexoEstudiantes().

Ejercicio 8

a) Crea una clase llamada StringUtil para trabajar con cadenas de caracteres de tal manera que no debas instanciarla para poder utilizarla. Esta clase debe contener los siguientes métodos:

- Concatenar (frase1, frase2): concatena a frase1 el contenido de frase2. El método debe devolver la frase resultante.

- BuscaYElimina (frase, palabra): busca la palabra en la frase y la suprime de la misma. El método debe devolver la frase resultante o avisar si la palabra no se encuentra en la misma mediante una excepción.

b) Crea un proyecto WPF para testear los métodos que has implementado, con la siguiente interfaz: