

Práctica – 2ª Ev. – 1ª Parte: Arrays, objetos y estructuras

Normas de obligado cumplimiento

Por cada fichero que cree o utilice se incluirá el siguiente comentario inicial:

```
/*  
  Autor: XXXXX  
  Fecha creación:      dd/mm/aaaa  
  Última modificación: dd/mm/aaaa  
  Versión: x.xx  
*/
```

Para cada función o método que creéis, incluiréis la siguiente información en un comentario:

- Descripción de lo que realiza función
- Parámetros de Entrada, Salida o Entrada/Salida
- Valor que retorna la función, si procede.

Para documentar los métodos utilizaremos las características que nos proporciona el lenguaje C# y el entorno de desarrollo Microsoft Visual Studio.

```
/// <summary>  
/// Descripción del proposito de la función o método  
/// </summary>  
/// <param name="XXXX">Descripción del parametro, para que se  
usa</param>  
/// <returns>Significado del valor devuelto por la  
función</returns>
```

Para utilizar la funcionalidad de documentar un programa, tan solo tenéis que comenzar un comentario con /// en la línea o líneas precedentes a la declaración de una función. Utilizando este sistema lograréis que al utilizar dichas funciones el editor os muestre ayuda referente a ellas.

Dichos comentarios pueden ser más adelante utilizados para generar la documentación de la aplicación utilizando herramientas específicas.

Comentarios

Cada problema de los expuestos a continuación se creará en un proyecto independiente, que llamaréis “Problema 1”, “Problema 2”, etc.

Los problemas tienen como propósito obligaros a que trabajéis el lenguaje de programación así como que profundicéis en vuestros conocimientos sobre la metodología de la programación (como resolver los problemas). Esta práctica está pensada para que la realicéis en clase completando el trabajo en vuestra casa. Ante cualquier duda acerca de cómo afrontar un problema deberíais preguntar al profesor al respecto para que el os oriente.

Problema 1 – Operaciones con arrays y cadenas

Se desea realizar un programa de gestión de un hospital. El programa controlará los siguientes procesos relacionados con el hospital:

¿Cómo funcionará el sistema?

- **Urgencias:** Para cada paciente que llegue al hospital por urgencias, el usuario del programa tomará sus datos personales, le asignará un médico y un tratamiento. Si es nuevo, se tomarán sus datos además de ponerle tratamiento (e ingresarlo, si es necesario). Si ya está en el sistema, se le pondrá tratamiento y se le podrá ingresar. En caso de ingreso, se modificará su estado (alta o ingreso).
- **Hospitalizaciones:** Cada paciente que se encuentre en el estado “ingreso” accederá a una cama hospitalaria libre. Para ello, el programa mostrará las camas del hospital y el usuario del programa le asignará una cama libre al paciente (esta información se guardará junto con los datos del paciente). El médico asignado puede ser cambiado en el momento del ingreso, pero una vez asignado el ingreso, no podrá modificarse mientras dure éste. El tratamiento podrá ser aplicado en cualquier momento, debiendo quedar constancia de cada tratamiento aplicado y la fecha/hora de aplicación. Una vez que el paciente se haya recuperado, recibirá el “alta” y volverá a dejar en estado “libre” su cama, quedando guardado en su historia su último médico asignado y la lista de tratamientos recibidos, junto con la fecha/hora de cada uno.
- **Consultas Externas:** Cada paciente que se encuentre en el estado de “alta” y tenga médico asignado podrá acceder a este servicio. En el programa, se le asignará un nuevo tratamiento, en una fecha/hora concreta, pudiéndose en este momento cambiar el médico asignado con anterioridad.

¿Qué datos hay que guardar?

- **Paciente:** aquí se almacenará la mayoría de la información del programa. Se deben guardar los datos del paciente en una lista de pacientes, ordenados por orden de llegada, con un número generado automáticamente de HISTORIA CLÍNICA, al darlo de alta en el sistema. Se guardará además: DNI (número único en la lista de pacientes), NOMBRE COMPLETO, FECHA DE NACIMIENTO, SEXO, DIRECCIÓN y TELEFONO. Se guardará, en cada momento, el ESTADO en el que se encuentre, su CAMA_Asignada (sólo si está ingresado), su Último_MÉDICO_Asignado (si ha sido atendido alguna vez) y un T.A.D. LISTA DE TRATAMIENTOS que incluya: Número_de_TRATAMIENTOS, array de TRATAMIENTOS (tipo string) y array de FECHA/HORA (tipo DateTime). El T.A.D. tendrá las siguientes operaciones: Inicialización (asignamos 0 a la variable que guarda el nº de tratamientos), Inserción de un elemento (añadimos un tratamiento al final de un array y una fecha y hora al final del otro array), Recorrer la lista (para visualizarla, por ejemplo) y Buscar un tratamiento (por la fecha o por el nombre), para visualizarlo. Las operaciones que el programa permitirá hacer con los pacientes son: “Buscar Paciente por DNI”, “Nuevo Paciente”, “Visualizar Datos Personales”, “Visualizar Historia – Tratamientos”, “Modificar Datos Personales Paciente”, “Asignar Médico”, “Tratar Paciente”, “Ingresar Paciente” y “Dar de Alta Paciente”.

- **Camas:** se configurará el hospital como una matriz bidimensional, siendo las filas las camas por habitación y las columnas el número total de habitaciones. En cualquier momento se podrá solicitar

la visualización del hospital y la ocupación de una cama “libre”.

Una posible forma de almacenar en el Paciente su Cama_Asignada en un ingreso, es con dos enteros: cama (1 ó 2) y habitación (1 .. 7).

Las operaciones que el programa permitirá hacer con las camas serán: “Ingresar Paciente”, “Dar de Alta Paciente” (son las operaciones indicadas en el párrafo de los Pacientes, que interactuarán con las Camas), “Configurar habitaciones” -donde se indicará el número de habitaciones y camas por habitación del hospital- y “Visualizar habitaciones” -que podría mostrar algo como lo siguiente-

	Habit. 1	Habit. 2	Habit. 3	Habit. 4	Habit. 5	Habit. 6	Habit. 7
Cama 1	libre	ocupada	libre	libre	ocupada	libre	libre
Cama 2	libre	libre	ocupada	libre	ocupada	libre	libre

-Sobre los **médicos** no guardaremos más que el nombre en esta versión del programa. Las operaciones que se pueden hacer serán: “Asignar Médico” y “Tratar Paciente” (que se indican también en el párrafo de Pacientes).

Realizar la programación de dichos procesos, accesibles a través de Menús, para ejecutarlo en la consola del sistema.

Sugerencia de menús: hacer menús organizados por procesos relacionados con el hospital (urgencias, hospitalización, consultas externas), indicando en cada uno las operaciones disponibles según las especificaciones dadas. Filtrar los datos que se van a introducir. Realizar pruebas con distintas configuraciones de hospital y datos de pacientes.

Anexo I – Manejando proyectos con múltiples ficheros en Visual Studio

Véase ejemplo 5 del tema 5 para una introducción al respecto.

Anexo II – Estructuras: “struct”

MSDN - <https://msdn.microsoft.com/es-es/library/0taef578.aspx>

Definición y acceso a datos - <http://www.nachocabanes.com/csharp/curso/csharp04c.php>

Estructuras (y más) - <http://www.mundonet.es/estructuras-las-clases-en-profundidad-herencia-interfaces-ejercicio-6-ejercicio-7.html?Pg=Entrega13.htm>

Anexo III – Filtrado de textos / números con Expresiones Regulares

PDF:

http://jms32.eresmas.net/web2008/documentos/informatica/lenguajes/puntoNET/System/Text/Regex/C_Sharp_ExpresionesRegulares.pdf