

PRÁCTICA 3ª: Creación y definición de una clase para el manejo de horas.

OBJETIVOS: Concepto de clase, atributos (datos miembro) y métodos (funciones miembro). Funciones externas. Métodos **const**.

TEMPORIZACIÓN:

Publicación del enunciado: Semana del 30 de septiembre.

Entrega: Semana del 21 de octubre junto con las prácticas 4 y 5.

Límite de entrega (con penalización): Semana del 28 de octubre.

BIBLIOGRAFÍA

Programación orientada a objetos con C++

Autor: Fco. Javier Ceballos

Editorial: RA-MA.

Se debe implementar la clase especificada por:

```
class CHora
{
    private:

        int m_nHoras;
        int m_nMinutos;
        int m_nSegundos;
        char *m_pszFormato; // almacena los valores "AM", "PM" o "24"

        bool Formato24() const;
        // Devuelve true si está en formato 24 horas.
        // Devuelve false si está en formato 12 horas (AM o PM).

    protected:

        bool EsHoraCorrecta() const;
        // Verifica si una hora es correcta: formato correcto (24, AM
        // o PM) y horas, minutos y segundos dentro de los límites
        // establecidos según el formato.
        // Devuelve true si es correcta y false en otro caso.

    public:

        void Construir();
        // Será invocada cada vez que se defina un objeto.
        // Pone a 0 los datos int y asigna el bloque de memoria
        // (3 caracteres) que será referenciado por el puntero
        // m_pszFormato e iniciado con una cadena vacía.

        bool AsignarHora(int nHoras,
                        int nMinutos,
                        int nSegundos,
                        const char *pszFormato);
        // Recibe 4 argumentos, correspondientes a una hora,
        // y los almacena en las variables
        // m_nHoras, m_nMinutos, m_nSegundos y m_pszFormato
```

```
// correspondientes al objeto que recibe el mensaje.  
// Invoca a EsHoraCorrecta.  
// Devuelve el valor retornado por EsHoraCorrecta.  
  
void ObtenerHora(int& nHoras,  
                int& nMinutos,  
                int& nSegundos,  
                char *pszFormato) const;  
// Obtener una hora. Permite obtener los datos hora,  
// minutos, segundos y formato correspondientes  
// al objeto que recibe el mensaje.  
  
void Destruir();  
// Libera la memoria reservada dinámicamente para un objeto y  
// pone el puntero m_pszFormato a cero.  
  
};
```

Guarde la declaración de la clase en un fichero `CHora.h`, y la definición en `CHora.cpp`.

El formato de la hora, AM o PM, se almacenará en minúsculas pero se permitirá introducirlo en mayúsculas o minúsculas.

En prácticas posteriores aprenderá que las tareas realizadas por los métodos `Iniciar` y `Destruir` son tareas propias de los constructores y destructores de las clases, por lo que no será necesario implementarlos.

El programa principal se escribirá en un fichero `práctica3.cpp` y deberá mostrar el siguiente menú:

1. Introducir hora
2. Visualizar hora
3. Terminar

Para mostrar la hora se utilizará una función externa con el siguiente prototipo:

```
void VisualizarHora(const Chora& hora);
```

Utilice los ficheros `utils.h` y `utils.cpp` de la práctica anterior. Puede añadir a estos ficheros nuevas funciones o mejorar las existentes. Por ejemplo, es aconsejable añadir una función `LeerCadena` que, utilizando el método `basic_istream::getline`, permita leer una cadena de una determinada longitud o hasta encontrar un carácter delimitador (por ejemplo `'\n'`); otra función puede ser `Minúsculas` que convierta el contenido de un objeto `string` en minúsculas.

Realizar otra versión del programa utilizando el tipo `string` en lugar de `char *`. No implemente los métodos de `CHora` anteriormente descritos que no sean necesarios al utilizar el tipo `string`.