

## PRÁCTICA 5ª: Construcción de una clase CFicha.

**OBJETIVOS:** Miembros de una clase que son a su vez objetos de otra clase.

### TEMPORIZACIÓN:

**Publicación del enunciado:** Semana del 25 de octubre.

**Entrega:** Semana del 8 de noviembre junto con las prácticas 3 y 4.

**Límite de entrega (con penalización):** Semana del 29 de noviembre.

### BIBLIOGRAFÍA

**Programación orientada a objetos con C++**

Autor: Fco. Javier Ceballos

Editorial: RA-MA.

---

Se construirá una clase CFicha con la siguiente funcionalidad:

- Atributos privados:

```
string m_sNombre;  
int m_nEdad;  
CHora m_horaNacio; // OJO: clase CHora de la práctica 4 versión  
// string
```

- Métodos:

Constructor: tendrá como argumentos por defecto una cadena vacía ("") para el nombre, 0 para la edad y 12:00:00 AM para la hora:

```
CFicha::CFicha(const string& sNombre, int nEdad, int nHoras,  
int nMinutos, int nSegundos, const string& sFormato)
```

En la definición del constructor utilizar una lista de iniciadores.

Método AsignarNombre. Este método copiará en el dato m\_sNombre del objeto el nombre que se le pase como argumento:

```
void CFicha::AsignarNombre(const string& sNombre)
```

Método ObtenerNombre. Este método devolverá una cadena copia del dato m\_sNombre del objeto:

```
string CFicha::ObtenerNombre() const
```

Método AsignarEdad. Este método copiará en el campo m\_nEdad del objeto el valor que se le pasa como argumento:

```
void CFicha::AsignarEdad(int nEdad)
```

Método `ObtenerEdad`. Este método devuelve el dato `m_nEdad` del objeto:

```
int CFicha::ObtenerEdad() const
```

Método `AsignarNacio`. Este método copiará la hora, que introduzca el usuario, en el campo `m_horaNacio` del objeto. Para ello utilizará el método `AsignarHora`, miembro de la clase `CHora`. Devuelve lo mismo que `AsignarHora`:

```
bool CFicha::AsignarNacio(int nHoras, int nMinutos, int  
                          nSegundos, const string &sFormato)
```

Método `ObtenerNacio`. Este método devuelve un objeto de la clase `CHora` con los valores del dato miembro `m_horaNacio`:

```
CHora CFicha::ObtenerNacio() const
```

Para la realización de la práctica se han de utilizar los ficheros `CHora.h` y `CHora.cpp` realizados en la práctica 4 (versión con `string`). La declaración de la clase `CFicha` se realizará en un fichero `CFicha.h` y su definición en un fichero `CFicha.cpp`. El programa principal se escribirá en un fichero `práctica5.cpp` y mostrará un menú como el siguiente:

1. Crear una ficha (*La ficha será creada dinámicamente*)
2. Visualizar ficha
3. Terminar

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

```
void VisualizarFicha(CFicha& ficha);
```

Para reutilizar código ya escrito, esta función invocará a `VisualizarHora`.

Para obtener los resultados que se piden a continuación es necesario que la clase `CHora` tenga implementados explícitamente el constructor copia, el operador de asignación y el destructor.

Edite una función `main` como la siguiente (para esta prueba puede comentar, `/* ... */`, en lugar de eliminar, el código actual de la función `main`):

```
int main()  
{  
    CFicha ficha1;  
    CHora hora1;  
    hora1 = ficha1.ObtenerNacio();  
}
```

Verifique que el método `ObtenerNacio` de `CFicha` tiene el prototipo siguiente:

```
CHora CFicha::ObtenerNacio() const
```

Ponga un punto de parada en la sentencia sombreada y ejecute el programa paso a paso (*Depurar* → *Iniciar depuración* → *F10/F11*). Anote los métodos que se ejecutan hasta finalizar la ejecución de la sentencia sombreada.

Vuelva realizar el mismo proceso modificando `ObtenerNacio` como se indica a continuación:

```
CHora& CFicha::ObtenerNacio()
```

Observar que ahora devuelve una referencia y no es `const`.

¿Qué métodos no se han ejecutado ahora con respecto a la prueba anterior? ¿Cuál es la explicación? La respuesta depende de que `ObtenerNacio` devuelva un objeto temporal o una referencia.