



PRÁCTICA Nº 2: 1 sesión

(del 21 al 27 de Marzo de 2002)

Tipos Abstractos de Datos. Utilización e implementación de clases

0. OBJETIVO

El objetivo de esta práctica es tanto la utilización de una clase dada realizando un programa que la utilice, como la implementación de una clase, que debe compilar con un programa dado.

1. TIPO ABSTRACTO DE DATOS

Un tipo abstracto de datos (TAD) es, básicamente, una colección de valores junto con unas determinadas operaciones definidas sobre ellos. Para el usuario del tipo abstracto de datos sólo importan los valores que puedan guardar las variables asociadas al TAD y las operaciones de manipulación del mismo, dejando para el programador la tarea de la implementación de la estructura o tipo de dato que soporte el TAD y de las operaciones relacionadas con él.

En esta práctica vamos a realizar los dos papeles: Por un lado el papel de usuarios de un tipo abstracto de datos (tipo *Conjunto de enteros*); y por otro el papel de programadores de un tipo abstracto de datos (Tipo *Fechas*.)

2. TAREAS A REALIZAR

Utilización de la clase *Conjunto de enteros* para implementar la criba de Eratóstenes

Como ejemplo de utilización de la estructura de datos *Conjunto de enteros*, se propone escribir un programa que implemente el método de obtención de números primos propuesto por Eratóstenes y conocido como la criba de Eratóstenes aplicado a los 'N' primeros números naturales.

El método consiste, básicamente, en los siguientes pasos:

- 1.- Pedir el número 'N' al usuario.
- 2.- Guardar los números que queremos cribar en un *Conjunto* (desde 2 hasta N),
- 3.- Desde i con valor 2 hasta N
 - a.- Si el elemento 'i' está en el conjunto, 'i' es primo (mostrarlo por pantalla.)
 - b.- Eliminar del conjunto todos aquellos elementos que sean múltiplos de 'i' (incluyendo el propio 'i'.)

La clase 'Conjunto de enteros' tiene la siguiente interfaz:

```
class Conjunto
{
    public:
        void IniciarConjunto (void);
        bool AnadirElemento (int);
        bool BorrarElemento (int);
        bool EsConjuntoVacio (void);
        bool Pertenece (int);

    private:
        .??
};
```

Recordad que la parte **private** es cosa del programador, y no nos importa a la hora de utilizar la



clase como usuarios.

Tanto la interfaz (**conjunto.h**), como la implementación de la clase (**conjunto.o**) será proporcionado por el profesor de prácticas (y estará en la página de la asignatura: <http://informatica.uv.es/iiguia/AED>)

Implementación de la clase *Fecha*

Crearemos una clase 'fecha' con la siguiente interfaz:

```
class Fecha
{
    public:
        bool IniciarFecha (int dia, int mes, int anyo);
        void DeterminarFecha (int & dia, int & mes, int & anyo);
        void EscribirFecha (void);
        bool IgualFecha (fecha);
        bool MayorFecha (fecha);
        bool MenorFecha (fecha);

    private:
        .??

};
```

En este caso, como somos los programadores de la clase, tendremos que decidir cual es nuestra parte privada y a partir de esta decisión implementar los diferentes métodos de la clase.

Esta clase deberá funcionar con el programa '**OrdenarFechas.cpp**' proporcionado por el profesor. El programa '**OrdenarFechas.cpp**' lee fechas de un fichero, y las muestra ordenadas por pantalla, y no puede ser modificado en ningún punto.

3. REQUISITOS

Recordar que para poder acceder a la sesión de prácticas correspondiente es necesario entregar un pequeño esquema de lo que se va a realizar en la práctica. En esta práctica, habrá que entregar dos documentaciones de programa. Una completa para la criba de Eratóstenes, explicando el programa principal y la clase '*Conjunto*', y otra en la que sólo vamos a explicar la clase '*Fechas*' (el programa principal estará hecho por el profesor.) Si no se presenta la documentación ANTES de acceder al laboratorio, no se permitirá la entrada a la sesión de prácticas.

4. ENTREGA

La entrega de esta segunda práctica debe incluir **el código** con el programa principal y las funciones del programa que implemente la criba de Eratóstenes (**criba##.cpp**), y la implementación de la clase 'Fecha' (**fecha##.cpp** y **fecha##.h**)

La entrega puede realizarse por e-mail a la dirección del profesor/a encargado@ del grupo o en disquette al iniciar la siguiente sesión de prácticas, pero siempre ANTES de empezar la siguiente sesión de prácticas.

Cualquier práctica entregada fuera de plazo no será admitida para su corrección.



Fecha límite de entrega de prácticas: Sesiones del 10 al 12 de Abril