

Notas:

- 1.- La definición de la clase se debe realizar en un .h, mientras que la implementación de la misma se debe realizar en el punto cpp, excepto se solicite lo contrario.
 - 2.- Los ejercicios son evolutivos y dentro de esta serie, un ejercicio depende del ejercicio anterior.
 - 3.- El código debe ser siempre el menor posible. Esto significa que, si una porción de código se repite, realizar una macro o encapsularlo en una función o método. Analizar si la codificación realizada es realmente necesaria.
 - 4.- Siempre que sea posible, aunque el enunciado no lo exprese, utilizar el atributo **const**.
 - 5.- Más allá del carácter evolutivo de la serie, en cada punto desarrollar un main.cpp que permita realizar una prueba mínima sobre la clase.
-

1. Definir e implementar la clase Dia, para el manejo de fechas.

La clase debe poder manejar fechas en el modo dd/mm/aaaa, para lo cual se considera de debe tener al menos los siguientes elementos públicos y privados:

Variables miembro no públicas:

- **dd y mm**- del tipo **char**, que representan el día y mes respectivamente de un determinado momento.
- **aa**- del tipo **short**, que representa el año de un determinado momento.

Métodos miembro no públicos:

- **verifica_dia**- recibe los valores correspondientes a un determinado día (dd,mm y aa) e informa si la fecha es o no válida.

Métodos miembro públicos:

- **setDia, setMes, setAnio**: permiten configurar en forma independiente cada una de las variables miembros privadas. La acción se debe concretar siempre y cuando la fecha exista. El método retorna **true** o **false** según corresponda.
- **getDia, getMes, getAnio**: permiten obtener en forma independiente cada uno de los parámetros.
- **set**: permiten configurar el día, mes y año. La acción se debe concretar siempre y cuando la fecha exista. El método retorna **true** o **false** según corresponda.

Métodos públicos de clase: (no corresponden a un objeto)

- Implementar un método de la clase, que al pasarle los tres parámetros (día, mes y año), indique si la fecha existe o no.

Sobrecarga de operadores públicos:

- **operator++**: incrementa la fecha en uno. Considerar pos y pre incremento.

- **operator--**: decrementa la fecha en uno. Considerar pos y pre decremento.
- **operator+**: permite sumarle a una fecha una determinada cantidad de días.
- **operator-**: permite descontarle a una fecha una determinada cantidad de días o bien indicar la cantidad de días que hay entre dos fechas, según corresponda.
- **operator<<**: imprime la fecha en formato dd/mm/aaaa.

Constructor:

- Solo con los tres parámetros (día, mes y año) en forma explícita. Si la fecha seleccionada no existiese, se debe configurar la fecha próxima siguiente o anterior según se defina en el proceso de diseño.

Por ejemplo, se configura 40/5/1998, se podrá crear el objeto Dia como 1/6/1998 o bien como 31/5/1998, según se defina.

Notas:

- *Considerar años bisiestos.*
- *Considerar fechas según el calendario actual. Omitir el análisis histórico de cambios en el calendario.*

2. Definir e implementar la clase Fecha, la cual hereda en forma pública de Dia y agrega las siguientes funcionalidades.

Variables miembro y de clase, públicas y no públicas:

- Agregar las variables que considere necesario para poder implementar los métodos y acciones que se requieren a continuación.

Métodos públicos de clase: (no corresponden a un objeto)

- Implementar un método de la clase, que permita definir el idioma en que se muestren los días de la semana y los meses. Implementar al menos las opciones en español e inglés.
- Implementar un método de la clase, que permita definir el formato en que se imprimirá una fecha al usar la sobrecarga de salida de flujo. Implementar al menos 5 formatos, de los cuales al menos 3 de ellos incluyan, el nombre del día de la semana y/o nombre del mes en el idioma configurado.

Métodos miembro públicos:

- **getsDia, getsMes**: devuelven respectivamente en formato texto (**string**) el nombre del día de la semana (lunes, martes, ...) y del mes (enero, febrero, ...), en el idioma que este configurada la clase.
- **getxDia**: devuelve en número de día de la semana (queda a criterio del desarrollador, definir la correlación número – día de la semana)
- **addDias, addMes, addAnio**: permiten sumarle a una fecha una determinada cantidad de días, meses o años según corresponda. En caso de ser necesario, el resultado de debe ajustar a una fecha valida.
- **subDias, subMes, subAnio**: permiten substraerle a una fecha una determinada cantidad de días, meses o años según corresponda. En caso de ser necesario, el resultado de debe ajustar a una fecha valida.

Sobrecarga de operadores públicos:

- Agregue los operadores de comparación: == != > >= < y <=.

3. Definir e implementar la clase Hitos, la cual permita guardar un acontecimiento pasado o futuro, según se la quiera implementar.

La clase debe contener al menos los siguientes campos no públicos relacionados con el hito:

- Título, del tipo **string**
- Fecha, del tipo **Fecha**
- Descripción, del tipo **string**
- Nivel de importancia, del tipo **int**

Queda a criterio del equipo de desarrollo los componentes (métodos, variables, constructores, etc) necesarios para la implementación de la clase.

4. Definir e implementar una lista utilizando la Standard Template Library (STL), que permita almacenar datos tipo Hitos, de manera ordenada por Fecha. En caso de Fechas coincidentes, definir los mecanismos que determinen el orden en el que se almacenaran los datos.

La lista deberá permitir como mínimo:

- Insertar elementos en forma ordenada.
- Extraer elementos desde el principio o desde el final.
- Extraer elementos dada una determinada Fecha o un determinado Título. En caso de haber más de un elemento que coincida con lo solicitado, indicar el criterio con el cual se extraerá el elemento.
- Obtener elementos desde el principio o desde el final (sin eliminar de la lista).
- Obtener elementos dada una determinada Fecha o un determinado Título. En caso de haber más de un elemento que coincida con lo solicitado, indicar el criterio con el cual se obtendrá el elemento (sin eliminar de la lista).
- Imprimir todos los elementos de la lista.
- Imprimir los elementos que coincidan ciertas características (al menos Fecha y/o Título)

Notas:

- *La lista no debe permitir la inserción de dos elementos iguales (usar iteradores)*
- *Se recomienda el uso de iteradores para recorrer la lista*
- *Revise la documentación y analice los ejemplos desarrollados en los enlaces que se detallan mas abajo.*

En los siguientes enlaces podrán encontrar información respecto a este Template u otros que pudiesen considerar más apropiado.

<https://www.geeksforgeeks.org/list-cpp-stl/>
<https://www.cplusplus.com/reference/list/list/?kw=list>
