

PRÁCTICA 7

1. DISEÑAR

1) Defina atributos y métodos para los siguientes objetos (para una funcionalidad mínima).

- a. Fecha.
- b. Automóvil.
- c. Pava eléctrica.
- d. Ventana del sistema operativo Windows.
- e. Sala de cine.

2. DISEÑAR E IMPLEMENTAR

2) Diseñe e implemente en C++ los siguientes objetos (creando clases). En cada caso incluya prototipos para un constructor (o varios) y funciones miembro que permitan manipular cada uno de sus atributos. Por ejemplo, si la clase tiene el atributo `hora`, implementar las funciones `setearHora(unaHora)` y `mostrarHora()`. Esto hace que se puedan manipular de forma correcta los miembros de datos privados de la clase. Implemente las funciones pedidas e incluya otras funciones si es necesario.

- a. Una clase llamada `Fecha` con atributos de tipo entero llamados `dia`, `mes` y `año`.
- b. Una clase llamada `Complejo` que tenga miembros de datos (atributos) en número de precisión doble llamados `real` e `imaginario`.
- c. Una clase llamada `Circulo` que tenga miembros de datos en número entero llamados `centro_x` y `centro_y` y un miembro de datos en número de precisión doble llamado `radio`.

3) Diseñe e implemente en C++ una clase llamada `Rectangulo` que tenga atributos en número de precisión doble llamados `base` y `altura`. La clase deberá tener una función miembro llamada `perímetro()` y otra función miembro llamada `area()` para calcular el perímetro y el área de un rectángulo. Implementar las funciones para manipular los atributos (asignarles valor y mostrar). Además, implementar una función miembro llamada `mostrarDatos()` que muestre la base, altura, perímetro y área de un rectángulo (utilizando los propios métodos de la clase).

4) Diseñe y luego implemente en C++ una clase llamada `Estudiante` consistente de un número entero de identificación del estudiante, un arreglo de 60 calificaciones (enteros) y un entero que represente el número total de calificaciones introducidas. El constructor de esta clase deberá inicializar todos los miembros de datos de `Estudiante` en 0. Deberán incluirse en la clase funciones para 1) introducir un número de identificación del estudiante, 2) introducir una calificación (si no fueron introducidas ya todas las notas) y 3) calcular la calificación promedio y mostrar la identificación del estudiante seguida por la calificación promedio.

5) Modifique la clase `Fecha` (ejercicio 2) a) para que contenga un método que compare dos objetos de tipo `Fecha`. El método lo recibe un objeto de clase `Fecha` y tiene como parámetro otro objeto de clase `Fecha`. Retorna verdadero si la fecha es menor a la `Fecha` argumento:

```
fecha1.Menor (fecha2)
```

El método puede escribirse de acuerdo al siguiente algoritmo:

Algorithm 1: Función de comparación entre fechas.

```
1  acepta dos objetos Fecha como parámetro
2  determina la fecha más reciente mediante el siguiente procedimiento:
3      convierte cada fecha en un valor entero de la forma aaaammdd
4      (esto puede lograrse usando la fórmula año*10000 + mes * 100 + día)
5      compara los enteros correspondientes para cada fecha
6      el número entero más grande corresponde a la fecha más reciente.
7  devuelve la fecha más reciente
8  }
```

6) Diseñar e implementar en C++ la clase pila de enteros.

7) Diseñar e implementar en C++ la clase cola de enteros.

8) Diseñar e implementar en C++ una plantilla (template) para la clase pila.

9) Diseñar e implementar en C++ una plantilla (template) para la clase cola.

Nota: para compilar de forma correcta incluya el .h y el .cpp de la plantilla en el .cpp donde esté la función main(). Por ejemplo, en el .cpp donde se encuentra la función main() y se usan las funciones del template cola, se incluyen:

```
#include "cola.h"
#include "cola.cpp"
```