

## PROGRAMACIÓN II - GRUPO de Prácticas D

### PRÁCTICA Nº2

4 de marzo 2024 18:30-21:30

#### INSTRUCCIONES

---

1. Todos debéis acceder a la **SALA** del campus virtual que tenéis disponible.
2. Durante el desarrollo de la sesión práctica debéis seguir el procedimiento de trabajo en la sala para la realización de las actividades evaluables. Recordad que es **imprescindible GRABAR toda la sesión y COMPARTIR la pantalla completa**.
3. La entrega de la práctica sólo se admitirá a través de la **actividad** disponible en el campus virtual de la asignatura de **Prácticas de Programación II antes de la hora de finalización de la sesión de prácticas**.
4. Se recomienda entregar **un único fichero** en **formato .txt o .cpp, sin comprimir**, con el nombre de ***EjercicioPRII\_2\_GrupoD***. *Pero se admite entregar cada ejercicio en archivos separados*.
5. El fichero entregado debe **incluir el nombre de los integrantes del equipo, como código comentado en la primera línea del archivo**.
6. Aunque las prácticas se realizan en grupos de dos integrantes, para su evaluación, **ambos deberán hacer la entrega a través de su campus virtual**. En otro caso, la práctica quedará **sin evaluar** y supondrá un **0 en su calificación**.
7. **El incumplimiento de alguna de las instrucciones sobre la realización/entrega de la práctica supondrá su descalificación inmediata (0 puntos)**.

Al final del enunciado de la práctica tenéis la puntuación de los diferentes hitos que debéis cumplir durante la sesión. Se recomienda que realicéis de forma previa al desarrollo una lectura comprensiva del enunciado, así como un debate en cada equipo con el que podáis planificar la estrategia a seguir durante la práctica.

Dicha estrategia deberá quedar plasmada en un flujograma que deberá ser entregado en papel al finalizar la práctica, indicando en los nombres de los integrantes del equipo.

Recordad que podéis/debéis utilizar el chuleterio durante la sesión, y que debéis tener en cuenta la rúbrica general de evaluación. Por ser una práctica podéis consultarla durante la sesión sin problema (está disponible en el campus virtual de la asignatura).

#### **EJERCICIO PRÁCTICO [90% puntuación] – 180minutos**

---

En esta práctica vamos a continuar con una de las dos clases desarrolladas en la primera práctica. Podéis tomar la solución propuesta de la práctica 1 como base considerando las modificaciones del siguiente ejercicio

Clase *TermometroEuropeo*:

- Atributos [1 punto]:
  - **LecturaCentigrados** (variable del tipo que consideres oportuna para almacenar un número con dos decimales)
  - **LecturaMinima** (variable del tipo que consideres oportuna para almacenar un número con dos decimales)
  - **LecturaMaxima** (variable del tipo que consideres oportuna para almacenar un número con dos decimales), se recomienda que LecturaCentigrados, LecturaMinima y LecturaMaxima sean del mismo tipo.
  - **Descripción** (variable del tipo que consideres oportuna para almacenar una descripción de lo indicado por la lectura. Por ejemplo, la temperatura en Madrid)
- Miembros [1 punto]:
  - **Métodos getter** para todos los atributos de la clase, no tienen parámetros.
  - **Métodos setter** para los atributos LecturaCentígrados y Descripción
  - **Método GetFahrenheit** para obtener la temperatura equivalente en grados Fahrenheit usando la fórmula  $F = (LecturaCentigrados * 9) / 5 + 32$
- Requisitos clase TermometroEuropeo [1 punto]:
  - No debemos permitir que la lectura del termómetro marque menos de LecturaMinima ni más de LecturaMáxima. El setter del atributo LecturaCentígrados debe lanzar una excepción si no se cumplen estas dos condiciones. El mensaje de la excepción describe lo ocurrido (el valor no está dentro del rango admitido).
  - Por ejemplo:
    - `throw std::runtime_error("mensaje")`
- Constructores y destructor [3 puntos]
  - Un constructor por defecto (sin parámetros) que asigne 0 a LecturaMinima y otro para LecturaMaxima.
  - Un constructor que admita 2 parámetros, que serán 2 valores, uno para asignar a LecturaMinima y otro para LecturaMaxima.
  - Constructor de copia, copiará los 4 atributos privados descritos arriba
  - Declarar un destructor, el destructor muestra por pantalla el mensaje “se ha invocado el destructor”
- A desarrollar en el método main [3 puntos]:
  - Un array de dos posiciones que contenga elementos de tipo TermometroEuropeo.

- En la primera posición habrá un elemento que se ha inicializado usando el constructor para iniciarlo. Invocar el constructor con valores `LecturaMinima = -1` y `LecturaMaxima = 1`
- No es necesario que los 2 puntos anteriores se hagan en la misma línea de código.
- Invocar al setter de Descripción para asignar una descripción. Esta descripción es un texto libre.
- En la segunda posición, crear otro elemento, pero esta vez, asignarle los valores del elemento en la primera posición mediante el constructor de copia
- Al elemento de la segunda posición: Invocar al setter de `LecturaCentigrados` para todos los valores enteros entre  $-2$  y  $2$ . (hay más de una forma de hacer esto).
- No dejes excepciones sin capturar (bloques try-catch).
  - Como los valores  $-2$  y  $2$  no son válidos, se debería lanzar excepción (si cumple con los requisitos) con un mensaje indicando el motivo.
  - En el bloque catch, mostrar un mensaje de lo ocurrido. El mensaje tiene que incluir la descripción y los valores mínimo y máximo del termómetro (atributos de la clase)
- En ningún caso tiene que pedir al usuario que introduzca valores por consola

Recuerda que:

- Todos los atributos (variables miembro) deben ser privados
- Se permite el uso de variables y métodos privados de uso auxiliar, que el enunciado no lo exige, pero sean de utilidad para vuestro código.
- En el nivel de acceso público de la clase deben estar los métodos que queramos ofrecer (interfaz pública de la clase)
- El constructor por defecto es el constructor que no recibe ningún argumento
- Los constructores de una clase son unos métodos particulares que no “devuelven nada”, se llaman de la misma forma que la clase y se pueden sobrecargar (puede tener varios constructores que reciban distintos argumentos)
- Debes poner “los const & en su sitio...”, recuerda la rúbrica de evaluación general y la penalización al 40% que supone no hacerlo bien...
  - Paso por referencia constante en argumentos de tipo no simple
  - Retorno de valores de tipo no simple que “ya existan”
  - Los métodos que no modifiquen el estado de la clase deben ser declarados como métodos constantes.

### **RÚBRICA DE CALIFICACIÓN**

---

- El 10% de la nota de la práctica, se obtendrá sobre trabajo realizado durante la sesión (resolución de preguntas, dominio de la materia, etc.). Esta nota será individual, y se considerarán para su valoración aspectos como:
  - **Defensa individual** a cuestiones que plantee el profesor durante la práctica.
- El 90% restante será sobre la puntuación del trabajo entregado antes de la finalización de cada práctica (misma nota para cada integrante del equipo), siguiendo la siguiente rúbrica general para cada apartado puntuable:

	%máx. (*)
El elemento evaluable no compila o no se asemeja a lo que se pide	0%
El elemento evaluable no se aproxima suficientemente a lo pedido	40%
El elemento evaluable se aproxima suficientemente a lo pedido	60%
El elemento evaluable funciona correctamente y las estrategias y elementos de código elegido son adecuados.	100%

(\*) El porcentaje (% máx.) representa el valor máximo sobre el que se evalúa el elemento indicado.