

PROGRAMACIÓN II - GRUPO de Prácticas C

PRÁCTICA Nº1

22 de febrero 2024 12:30-15:30

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 debe entregar **un único fichero** en formato **.txt o .cpp**, **sin comprimir**, con el nombre de **P1EjercicioX_GC**.
5. El fichero entregado debe **incluir el nombre de los integrantes del equipo**.
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 él los nombres de los integrantes del equipo.

Recordad que podéis/debéis utilizar el **chuletarío** 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] – 180 minutos

En esta primera práctica vamos a desarrollar dos clases. A continuación, describiremos los requisitos que debéis tener en cuenta para cada una de ellas.

Clase ISBN: (4 puntos)

- Atributos: [0,75 puntos]
 - Código formado por 10 dígitos distribuidos en 4 campos que se indican a continuación:
 - **Identificador país:** variable del tipo que consideres oportuna para almacenar 2 dígitos con el valor 84 que representa España.
 - **Editorial:** variable del tipo que consideres oportuna para almacenar 5 dígitos.
 - **Nº ejemplar:** variable del tipo que consideres oportuna para almacenar 2 dígitos.
 - **Código de control:** variable del tipo que consideres oportuna para almacenar 1 dígito que se calculará en base a los anteriores atributos.
- Miembros:
 - **Métodos getter** para todos los atributos de la clase [0,25 puntos]
 - **Métodos setter** para todos los atributos de la clase. [0,5 puntos]
 - **Método calcularControl** para calcular el valor correspondiente al **dígito de control del ISBN** según se indica: Se multiplica cada uno de los nueve primeros dígitos por la posición que ocupan en la secuencia numérica, es decir, el primero por 1, el segundo por dos y así sucesivamente hasta el noveno que se multiplica por 9. Luego se **suman estas multiplicaciones** y el **resultado se divide entre 11**. Dicha división dejará un resto entre 0 y 10. Si el **resto está entre 0 y 9**, este mismo **valor es el del dígito de control**. Pero si el **resto es 10**, entonces se establece como dígito de control la letra **X**. [1,5 puntos]

Por ejemplo:

84	25340	22	5
----	-------	----	---

Cálculo del código de control:

$$8+8+6+20+15+24+0+16+18= 115$$

Luego, el código de control será 5

- Requisitos clase ISBN:
 - El valor inicial del país dentro del ISBN debe ser **84**. [0,5 puntos]
 - El código de control de ISBN se debe de calcular según lo indicado en el método **calcularControl** y tal como muestra el ejemplo.

- Se debe validar que el código de la editorial tenga **5 dígitos** y el número de ejemplar **2 dígitos** (Como aún no sabemos lanzar excepciones, si se detecta que el número de dígitos no es correcto se asignarán a cualquiera de esas variables el valor **0 para todos los dígitos**). [0,5 puntos]

Nota: El método `string to_string (int val)` permite convertir números enteros a cadenas.

Clase **Publicacion**: (6 puntos)

- Atributos: [1,5 puntos]
 - **Título**: (variable del tipo que consideres oportuna para almacenar un texto).
 - **Autoria**: (variable de **tipo contenedor** que sea capaz de almacenar **hasta 3** posibles autores del artículo con su nombre y apellidos).
 - Por ejemplo:

Luis Arias Perez
Maria Perez Amigo
 - **FechaPublic** (variable del **tipo contenedor** para almacenar la fecha de publicación con **dos campos numéricos** para almacenar los datos de la fecha: mes y año).
 - Por ejemplo:

2	2024
---	------
 - **NumPaginas** (variable que almacenará el número de páginas que tiene la publicación).
- Miembros:
 - **Métodos getter** para todos los atributos de la clase. [0,25 puntos]
 - **Métodos setter** para todos los atributos. [0,5 puntos]
 - **Método mostrarPublicacion** para mostrar el contenido de todos los atributos de una publicación por pantalla. Se debe evitar mostrar campos vacíos, por ejemplo, si la publicación solo tiene un autor, debe aparecer solo el nombre de dicho autor. [0,75 puntos]
- Requisitos clase Publicacion:
 - No debemos permitir que los objetos Publicacion que se implementen tengan **vacío** el atributo correspondiente a la **autoría**. Deberás incorporar en los métodos que realicen acciones sobre ese atributo algún mecanismo de protección que lo evite (Como aún no sabemos lanzar excepciones en el caso que detectéis esa posibilidad asignaréis el valor **“SIN DETERMINAR”** al primero de los autores) [1,5 puntos]

- No debemos permitir que el valor de la fecha de publicación no sea correcto. Luego, hay que validar que el mes esté comprendido entre 1 y 12. Respecto al año no se permitirá ningún valor menor a 1900 ni mayor del año actual. Deberás incorporar en los métodos que realicen acciones sobre ese atributo algún mecanismo de protección que lo evite (Como aún no sabemos lanzar excepciones en el caso que detectéis esa posibilidad asignaréis el valor 1 de 1900 a los campos que componen la fecha) **[1 punto]**
- Del mismo modo no debemos permitir que el número de páginas sea inferior a cero. En el momento que detectéis esa posibilidad deberéis asignar el valor **0** a la variable correspondiente **[0,5 puntos]**

Una vez que tengáis vuestras clases preparadas vamos a ver entre todos como podemos instanciar un objeto de cada una de ellas. Y que necesitaríamos para poder inicializar sus atributos (variables miembro).

Recuerda que:

- Todos los atributos (variables miembro) deben ser privados.
- 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.
 - **Trabajo en equipo.**

- 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 y particular para cada apartado puntuable:

Criterios generales de evaluación

	% máx. (*)
Funciones/Métodos: Si no se usa el paso por referencia constante cuando las variables de los parámetros de entrada no son de tipo simple.	40%
Tipos de datos y variables:	
• Uso de variables globales (fuera del ámbito de una función).	0%
• Si no se usan los tipos contenedor vistos en clase (std::array; std::vector; std::set; std::string, etc.) para las variables que lo necesiten.	0%
• Si no se usan punteros inteligentes (std::unique_ptr; std::shared_ptr) cuando sea necesario.	0%
• Si se utilizan punteros clásicos.	0%
No utilizar funciones/métodos cuando sea apropiado o se indique de forma explícita en el apartado/ejercicio de la actividad/examen.	40%
El programa no compila o no se asemeja a lo pedido.	0%
Si no se cumplen los criterios de entrega indicados en la actividad/examen.	0%
Si los apartados/ejercicios de la actividad/examen son resueltos de forma NO algorítmica , es decir, proponiendo una solución particular para unos determinados datos/valores en lugar de ser una solución general independiente de los datos/valores.	0%

Criterios particulares de evaluación

	% 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 elegidos son adecuados.	100%

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