

Los mil y un martes y jueves

Algoritmos y programación II - 1C2022

Trabajo práctico N°2

Grupal



Introducción

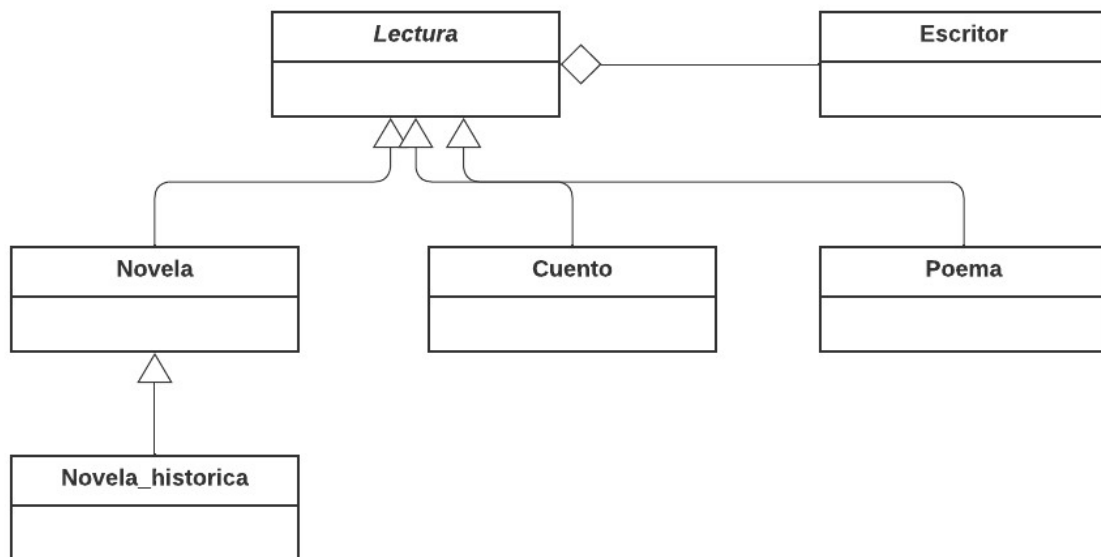
La ayuda que le dieron a Sid con el primer TP fue tan exitosa e incrementó de tal manera la cantidad de socios del club de lectura que les encargó una segunda tarea.

Los miembros del club se reúnen todos los martes y jueves a las 13hs. En ese momento eligen qué les gustaría leer en las tres horas siguientes. Las posibles lecturas deberán estar en una lista ordenadas por año, de menor a mayor. Las lecturas pueden ser novelas, cuentos o poemas. Además, las novelas pueden ser históricas, un género que consideraremos aparte.

Enunciado

La elección de una lectura puede determinarse de varias maneras que iremos viendo.

El diagrama de clases debe ser como en la siguiente figura:



La clase **Lectura** será abstracta, dado que nunca se leerá algo que no sea una novela, un poema o un cuento. Las clases tendrán los siguientes atributos, con sus métodos de obtener, no habrá métodos para cambiar los valores de los atributos, salvo excepciones, ya que se cargarán en los constructores.

- *Lectura* (clase abstracta)
 - título
 - minutos (minutos estimados que demandaría la lectura)
 - año (año de publicación)
 - autor (debe ser un puntero a un autor, con la finalidad de optimizar el espacio y de prevenir posibles inconsistencias).
- *Novela*
 - Género (enumerado que tiene: *DRAMA, COMEDIA, FICCION, SUSPENSO, TERROR, ROMANTICA, HISTORICA*)
- *Poema*
 - versos (cantidad de versos que tiene el poema)
- *Cuento*
 - Libro (título del libro en el que está publicado el cuento)
- *Historica*
 - Género (*HISTORICA*)
 - tema (Breve indicación sobre el tema histórico que aborda la novela, este campo debe ser un puntero a char)
- *Escritor*
 - nombres y apellidos
 - nacionalidad (puede ser '¿?')
 - año de nacimiento (-1 en caso de desconocerlo)
 - año de fallecimiento (-1 en caso de desconocerlo o de estar con vida, este es el único campo que puede cambiar)

Las lecturas estarán en un archivo de texto separadas por una línea vacía de donde se cargará la lista al principio de la aplicación. Además, habrá un archivo con los escritores. Estos archivos no tienen ningún orden en especial. La carga en la lista se hará ordenada por año, ante igualdad de año, el orden es indistinto.

Archivo escritores.txt

```
(número) --> número al que se hará referencia en el archivo de lecturas
Nombres y apellidos
Nacionalidad
Año de nacimiento
Año de fallecimiento
// línea en blanco
```

Ejemplo archivo escritores:

(1)
Julio Cortázar
argentino
1914
1984

(2)
Mary Shelley
británica
1797
1851

(3)
Stephen Edwin King
estadounidense
1947

(4)
Alfonsina Storni
argentina
1892
1938

(5)
Ken Follet
galés
1949

El archivo *lecturas.txt* tendrá la siguiente estructura:

N (novela), C (cuento) o P (poema)
título
minutos (estimados para su lectura)
año (de su publicación)
Si es novela, va el género
Si es novela histórica, va el tema
Si es cuento, va el título del libro en donde está publicado
Si es un poema, va la cantidad de versos
Referencia al autor (puede ser *ANONIMO*)

Ejemplo archivo lecturas:

C
Casa Tomada
20
1951
Bestiario
(1)

N
Rayuela
2900
1963
FICCION
(1)

N
Misery
2100
1987
TERROR
(3)

P
El Poema del Mio Cid
480
1140
3733
ANONIMO

P
Voy a dormir
1
1938
14
(4)

N
Frankenstein
1600
1818
TERROR
(2)

N
Una columna de fuego
4800
2017
HISTORICA
Guerra de religiones en Europa en la segunda mitad del siglo XVI
(5)

La aplicación debe leer estos dos archivos de entrada y cargar dos listas:

- escritores: manteniendo el orden del archivo: 1, 2, etc. Este valor no es necesario guardarlo.
- lecturas: ordenadas por año. Para hacer esto hay que implementar un método *comparar* que recibe una lectura y devuelve un int:
 - – 1 si la lectura que llama es menor a la que se pasa por parámetro
 - 1 si la que llama es mayor a la que se pasa por parámetro
 - 0 si son iguales
 - En el caso de autores anónimos, la lectura se cargará con un *null* en ese campo.
 - En el caso de una novela histórica, el tema debe guardarse de forma dinámica usando un *char**.

Las lecturas tienen que tener implementado el método *mostrar* para mostrar los valores de los atributos usando polimorfismo.

Luego de cargadas ambas listas, debe ofrecer un menú para:

1. Agregar una nueva lectura a la lista
2. Quitar una lectura de la lista
3. Agregar un escritor
4. Cambiar dato de un escritor (solo fecha de fallecimiento)
5. Listar los escritores
6. Sortear una lectura random para leer en una de las tertulias
7. Listar todas las lecturas (se debe implementar en Lectura el método *mostrar*)
8. Listar las lecturas entre determinados años. Los años los ingresa el usuario.
9. Listar las lecturas de un determinado escritor que ingresa el usuario.
10. Listar las novelas de determinado género.
11. Armar una cola ordenada por tiempo de lectura, de menor (van al principio) a mayor (van al final). Esta cola no debe duplicar las lecturas (cuidado a la hora de implementar el destructor). Para este punto deben sobrecargar el método *comparar*. Además, tienen que agregar una opción para indicar si una lectura fue leída, en ese caso, se saca de la cola.
12. Salir

Aclaraciones importantes

- **NO** se pueden utilizar las estructuras de STL.
- El trabajo deberá realizarse en equipo formado por tres integrantes.
- Deben entregar, además del código fuente, diagramas UML y documentación: manual de usuario y división de tareas.
- El trabajo debe compilar con los flags **-Wall -Werror -Wconversion**.

¿Qué se evaluará?

- Compilación: sin warnings ni errores.
- Funcionalidad.

- Eficiencia espacial.
- Eficiencia temporal.
- Buenas prácticas de programación (nombres descriptivos, sangrado, etc.).
- Separación de operadores.
- Modularización.
- Precondiciones y postcondiciones.
- POO, herencia y polimorfismo.
- Uso de la memoria dinámica.
- Diagramas UML y documentación.

Normas de entrega

Un integrante del equipo deberá subir al campus un único archivo comprimido en la sección TPs.

Este archivo deberá tener un nombre formado de la siguiente manera:

Nombre_equipo_TP2

El nombre del equipo puede tener una o dos palabras y no más de 15 caracteres en total.

Además, la documentación debe tener una carátula con el nombre del equipo y los nombres de los tres integrantes con sus respectivos números de legajo.

La fecha de **entrega** vence el **lunes 30 de mayo** a las 23.55hs.

Puntaje: 40 puntos.

Nota: este trabajo comprende una defensa oral (explicaciones de lo que hicieron, por qué lo hicieron de ese modo, etc.). Por lo que el puntaje puede no ser el mismo para todos los integrantes del equipo en caso de detectar que alguien trabajó menos o más.