

EXAMEN PARCIAL 2

Leé con cuidado el enunciado y por lo menos dos veces para resolver lo pedido. Pensá bien la estrategia de resolución antes de comenzar el desarrollo de lo que te solicitan. El objetivo de este examen es **evaluar la correcta aplicación de los conceptos y técnicas** vistos hasta el momento:

- Correcta definición de clases y asignación adecuada de sus responsabilidades.
- Encapsulamiento, ocultamiento de información y uso de getters y setters sólo cuando corresponda.
- Modularización reutilizable y mantenible con uso de métodos con correcta parametrización.
- Correcta aplicación de miembros de instancia y de clase.
- Correcta aplicación de herencia y polimorfismo.
- Correcta aplicación conceptual de las relaciones entre clases.
- Correcta aplicación de TADs vistas en clase.

Enunciado

Un Festival de Cine Independiente desea desarrollar un programa para acreditar a quienes deseen disfrutar de las 10 películas que se presentarán.

Las películas se proyectarán en una única sala con capacidad para 800 butacas, en tres **tipos de funciones** distintas:

- **Avant-Première** (una sola por película)
- **Función Privada** (una sola por película)
- **Función General** (tantas como hagan falta)

El sistema deberá procesar registraciones y, a partir de estas, guardar acreditaciones según el tipo de función y ciertas reglas que se mencionan posteriormente.

Registraciones

Las registraciones ya están **validadas y almacenadas** en el orden en que fueron recibidas y se procesarán en el mismo orden.

Cada registración incluye:

- DNI (entero)
- Nombre del interesado (texto)
- Nombre de la película que desea ver (texto)

Personas VIP

El Festival considera como VIP a ciertos participantes y mantiene una estructura con sus números de DNI. Se espera que el sistema pueda determinar, para cada registración, si el DNI del interesado figura entre los reconocidos como VIP.

Estos ya están cargados y tienen prioridad de acceso a las **dos funciones especiales** de cada película, que se mencionan a continuación.

Funciones y Nóminas

Cada película se identifica por su nombre y mantiene tres nóminas de acreditados, una por cada tipo de función:

- Las nóminas para cada una de las **dos funciones especiales** (Avant-Première y Función Privada): Cada una puede aceptar acreditaciones hasta completar la capacidad de la sala. Sus acreditaciones deben estar ordenadas por DNI.

EXAMEN PARCIAL 2

- La nómina para las funciones generales:
No tiene límite de acreditaciones.
Sus acreditaciones conservan el orden de registración.

Cada nómina debe saber cómo devolver una acreditación a partir de un **DNI específico**.

Reglas de Acreditación

Cada registración genera una acreditación para la película elegida y **solamente conserva el nombre y el DNI especificados en la registración**, ya que son los únicos datos requeridos para su identificación.

Al procesar cada registración se deben aplicar estas reglas:

- Si el DNI del interesado corresponde a una persona VIP:
 - Se intentará acreditarlo en la **Avant-Première**.
 - Si no hay lugar disponible, se lo intentará en la **Función Privada**.
 - Si tampoco hay lugar allí, se lo acreditará en la **Función General**.
- Si no es VIP:
 - Se lo acreditará directamente en la **Función General**.

Se requiere

Diseñar el diagrama UML completo que represente el problema, incluyendo clases, relaciones y estructuras necesarias para soportar las reglas planteadas.

El diseño debe considerar el uso de los **TADs vistos en clase**, según convenga.

Implementar en NS+ los siguientes métodos principales:

- `procesarRegistraziones()`
Procesa todas las registraciones. Cada una genera una acreditación y la ubica en la función adecuada de la película.
Al finalizar, no debe quedar ninguna registración pendiente.
- `obtenerAcreditadosPorTipoDeFuncionYPelicula()`
Devuelve una estructura que indica cuántos acreditados hay por tipo de función y por película.
- `obtenerAcreditacion(int)`
A partir de un DNI, devuelve la acreditación correspondiente si existe.
Se deben desarrollar también los métodos auxiliares necesarios.

Consideraciones adicionales

- Se espera un diseño orientado a objetos, con encapsulamiento y distribución adecuada de responsabilidades.
- Las estructuras internas deben usar los **TADs vistos en clase** (`ListaOrdenada`, `Cola`, etc.).
- El ordenamiento debe implementarse si se usan listas ordenadas.
- Se valorará el correcto uso de **composición**, **delegación** y **modularización**.