Universidad Da Vinci De Guatemala

Facultad de Ingeniería

Ingeniería en Sistemas

Logotipo

El contenido generado por IA puede ser incorrecto.

**Sistema de Reservaciones Para un Teatro**

Gustavo David Campos Morales

202400509

Estructuras de datos

Guatemala, marzo de 2025

**Explicación del Problema**

Eres el desarrollador de un sistema de reservaciones para un teatro y necesitas una estructura de datos que permita gestionar de manera eficiente la venta y cancelación de boletos. Debes diseñar un **Tipo de Dato Abstracto (TDA)** en Java que maneje las reservas de forma ordenada.

**Solución**

Vamos a crear un sistema de gestión de boletos para un teatro utilizando una cola y un conjunto ordenado de asientos disponibles. Vamos a crear la clase de Reservación que será la encargada de representar cada reservación e incluirá el nombre del cliente y el número de asiento asignado. Luego se procederá a crear otra clase que será la encargada de manejar la lista de reservaciones y los asientos disponibles. Permitirá agregar nuevas reservaciones, cancelar boletos y verificar si un asiento está ocupado. También tendrá una funcionalidad especial para asignar automáticamente el mejor asiento disponible, priorizando los más cercanos al escenario y por último se creará la clase principal Teatro que será la encargada de ejecutar el programa, este creará un gestor boletos, agregará algunas reservaciones manualmente y otras automáticamente, cancelará una reservación y mostrará la lista de boletos vendidos. Se usará una cola para almacenar las reservaciones en el orden en que fueron hechas, asegurando que se muestren en el mismo orden de llegada. También utilizaremos una estructura ordenada para gestionar los asientos disponibles, facilitando la asignación automática del mejor asiento libre.

**Código**



Texto

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de un celular

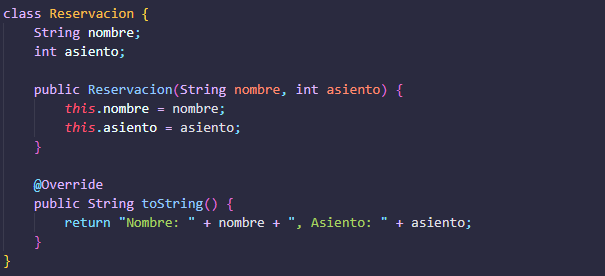
El contenido generado por IA puede ser incorrecto.

**Reservación**

**Texto

El contenido generado por IA puede ser incorrecto.**

Se importan las librerías necesarias para trabajar, en este caso se importan las librerías para poder trabajar con colas de tipo FIFO (First In, First Out) y para ordenar los datos de forma ascendente.



Se crea la clase Reservacion en donde tendremos 2 variables no inicializadas, una de tipo string para el nombre y una de tipo int para el número de asiento. Se crea un constructor con el nombre y asiento, luego se imprime la información de la nueva instancia del constructor.

**GestorBoletos**

Texto

El contenido generado por IA puede ser incorrecto.

Se crea un clase GestorBoletos y se crea una cola con el nombre reservaciones y un conjunto de datos que guarda los asientos disponibles, también se definen la cantidad de asientos en el teatro.

Texto

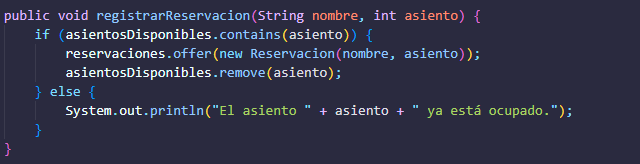
El contenido generado por IA puede ser incorrecto.

Se crea el constructor GestorBoletos que contendrá 2 parámetros; las filas y los asientos por fila, se crea una lista enlazada para las reservaciones y un conjunto de datos para los asientos disponibles.

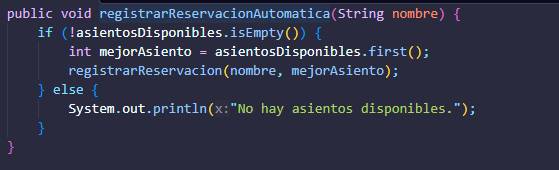
Texto

El contenido generado por IA puede ser incorrecto.

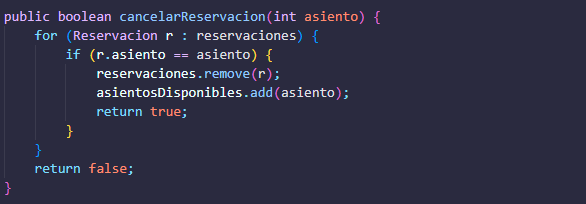
Se recorren las filas y los asientos para agregar cada número de asiento al conjunto de datos asientosDisponibles, se asume que los asientos están enumerados secuencialmente.



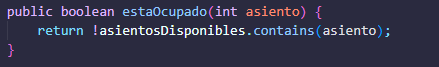
Con un condicional se verifica si el asiento está disponible, si está libre entonces se agrega la reservación a la cola y se elimina el asiento de asientosDisponibles, si no está libre, se muestra un mensaje diciendo que el asiento ya está ocupado.



Se verifica si hay asientos disponibles para crear una nueva reservación, este obtiene el asiento más cercado al escenario y da el menor número de asiento disponible asumiendo que los números bajos están más cerca de la pantalla. Llamaj a registrarReservacion y asigna una nueva reservación. Si no hay asientos disponibles muestra un mensaje diciendo que no hay asientos disponibles.



Busca en la cola de reservaciones si hay alguna con el asiento dato, si encuentra una coincidencia entonces la elimina de la cola, agrega el asiento nuevo a asientosDisponibles y retorna un dato booleano indicado éxito., si no, devuelve el dato booleano false.



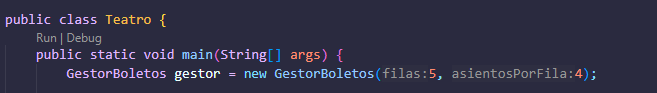
Se verifica si el asiento está ocupado, si el valor es true significa que el asiento no está en asientosDisponibles, si retorna false es porque sí está disponible.

Texto

El contenido generado por IA puede ser incorrecto.

Recorre la cola de reservaciones e imprime cada reservación.

**Teatro**



Se crea la clase Teatro y se crea una instancia en donde el valor de las filas es 5 y 4 asientos por fila.

Imagen de la pantalla de un celular de un mensaje en letras blancas

El contenido generado por IA puede ser incorrecto.

Se registra dos reservaciones con asientos específicos.

Una captura de pantalla de un celular con texto e imagen

El contenido generado por IA puede ser incorrecto.

Se registran dos reservaciones automáticas, asignando el mejor asiento disponible.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Imprime todas las reservaciones.



Muestra si el asiento 10 está ocupado



Cancela la reservación del asiento 10



Muestra las reservaciones después de la cancelación.

**Ejemplo de uso con capturas de pantalla o salida esperada**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Conclusión sobre lo aprendido**

En este caso como en la tarea anterior hemos utilizado una lista enlazada ya que nos permite agregar y eliminar elementos eficientemente, lo cual es perfecto para una reservación de tickets, también se aprendió a utilizar condicionales para manejar las listas, en este caso para verificar si estaban vacías o no para poder trabajar con ellas, también aprendimos a manejar colecciones de datos para almacenar en este caso las reservaciones y ordenar de forma ascendente los elementos, para asignar así números desde 1 hasta el límite total del lugar, con esto damos por entendido que los primeros números son los que tienen mejores asientos, ósea más cerca de la pantalla, entonces serán ordenados de está manera para ir sentando a las personas con un número de asiento más bajo, así cumplimos con el objetivo de sentarnos en el mejor lugar , se manejan las reservaciones, se eliminan también en caso de ser necesario y se actualiza la lista de las reservaciones.