



Universidad del Valle

Facultad de Ingeniería
Escuela de Ingeniería de Sistemas y Computación
Fundamentos de Programación Orientada a Objetos

LABORATORIO 1

Fecha de Entrega: 18 de septiembre de 2024

NORMAS PARA LA ENTREGA DE LOS LABORATORIOS

- Coloque el nombre de los integrantes del grupo (**EXACTAMENTE TRES ESTUDIANTES**), nombre del profesor, número del grupo, y número de laboratorio presentado en el encabezado **de todos sus archivos .cpp**.
- **Lugar y Medio de Entrega:** Todos los archivos que se soliciten en el informe se deben subir al campus virtual en el respectivo enlace del laboratorio.
- **Plazo:** Los estudiantes deben subir los archivos antes del día y la hora establecida por el profesor en el campus, el vínculo para esta actividad se deshabilitará automáticamente una vez se cumpla el plazo.
- **Durante el curso no se recibirán informes de laboratorio enviados por correo electrónico ni fuera de la hora de entrega especificada en el campus.**

DEBE DESARROLLAR EL LABORATORIO SÓLO CON LOS INTEGRANTES DE SU GRUPO DE TRABAJO, CUALQUIER COPIA CON OTRO GRUPO DE TRABAJO O COPIA DE CÓDIGO DE INTERNET CONSTITUYE FRAUDE POR ENDE LA ANULACIÓN DEL LABORATORIO PARA TODOS LOS GRUPOS IMPLICADOS

Punto 1: Modelando solución al problema

Se requiere una aplicación para gestionar la información de los diferentes vehículos que transitan por las calles de un municipio. La aplicación debe permitir el ingreso de:

- Cada vehículo,
- Cada municipio,
- Cada calle de los municipios y
- De las zonas que tienen restricciones de movilidad en las diferentes calles de los municipios.

También debe permitir :

- La búsqueda de Vehículos: por placa, por tipo de combustible, por capacidad de pasajeros
- Mirar por cuales calles de un municipio específico se puede circular en un horario determinado

- Listar los municipios ingresados, los vehículos y las calles de un municipio determinado.

Cada vehículo tiene una marca, una placa, un color, un modelo, un propietario, un tipo de motor, tipo de gasolina, cantidad de caballos de fuerza del motor, capacidad de pasajeros.

Los municipios tienen un nombre, los vehículos matriculados en dicho municipio, calles y tienen zonas por las que sus calles no permite la circulación de vehículos en unos horarios determinados y otros de manera permanente. Las zonas también tienen un nombre. Las calles tienen un nombre y el sentido de circulación.

Identifique los objetos, atributos y métodos necesarios para resolver el problema y realice:

- [0,9 ptos] El diagrama de clases para implementar la solución al problema antes planteado.

RECUERDEN QUE: en el diagrama de clases se debe especificar las clases con sus correspondientes atributos, métodos, tipos de datos y modificadores de acceso (encapsulación) y las relaciones que se presentan entre las clases. Para construirlo pueden hacer uso de Dia o cualquier otra aplicación de diagramación UML

- [0,6 ptos] La explicación de qué haría cada método de las clases que planteó en su diagrama de clases. Para ello puede hacer una tabla indicando: nombre del método, clase a la que pertenece, y descripción

Nombre del método	Clase a la que pertenece	Descripción - explicación del método

Punto 2 [3.5 ptos]: Cine FPOO

Cine FPOO cuenta con cinco salas de proyección, y requiere una aplicación para gestionar la información de sus salas y las películas que en ellas se proyectan.

*La aplicación debe permitir ingresar películas, e ingresar la información de las cinco salas con las que cuenta el cine. Cada sala tiene un nombre y la programación del día, en cada sala se proyectan 4 películas diarias. La programación se hace por película, por lo que tiene el nombre de la película, la hora de inicio y la hora de fin, las horas se debe leer por separado hora y minutos en formato de 24 horas. **Por ejemplo**, hora inicio: 11 minutos inicio: 30, hora fin: 13: minutos fin: 45.*

Cada película tiene un nombre, un género (comedia, drama, romance, acción, suspenso, ciencia ficción, terror, fantasía), y un precio de la boleta que se calcula de acuerdo con el género de la misma, de la siguiente manera:

- Las boletas de las películas de comedia y drama valen un 10% más del precio base.
- Las boletas de las películas de romance y acción valen un 15% más del precio base

- Las boletas de las películas de suspenso y terror valen un 18% más del precio base
- Las boletas de las películas de ciencia ficción y fantasía valen un 23% más del precio base

El precio base de las boletas se debe pedir cuando se inicie la aplicación.

La aplicación debe funcionar a través de un menú de opciones que contenga la siguiente:

1. **Ingresar película:** cada que se seleccione esta opción se debe solicitar los datos de una película y almacenarla, Se debe verificar que el género ingresado sea uno de los válidos, de no serlo se le debe indicar al usuario que no es posible ingresar la película.
2. **Ingresar Sala :** cada que se seleccione esta opción debe permitir el ingreso de la información completa de una de las cinco salas del cine (su nombre y su programación). Se debe verificar al ingresar las 4 programaciones de la sala que sea de una de las películas ingresadas anteriormente y que las horas sean válidas. (la sala de cine abre a las 7:30 y cierra a las 22:30) si se ingresa mal una programación se debe pedir al usuario los datos hasta que los ingrese de manera correcta. Al terminar se debe indicar al usuario que se ingresó la sala con éxito.
3. **Consultar películas por género :** Cada que se seleccione esta opción se debe solicitar un género (de los válidos) y se debe listar los nombres de las películas de dicho género, en qué sala se proyecta y a qué horas
4. **Consultar horario película:** Cada que se seleccione esta opción se debe solicitar el nombre de la película que se quiere consultar y debe listar en qué sala y en qué hora se proyectará.
5. **Salir:** Termina la ejecución del programa

Cada vez que se termina una opción, se debe mostrar nuevamente el menú principal. El programa sólo finalizará cuando se seleccione la opción salir.

Desarrolle la solución al problema antes planteado usando el enfoque orientado a objetos, debe entregar:

- **[1 pto]** Diagrama de clases que modele la solución al problema planteado (su solución debe tener al menos 4 clases)
- **[2.5 ptos]** La implementación de la solución al problema en C++.

La implementación del código se deberá hacer usando únicamente sintaxis del lenguaje C++, los arreglos permitidos para la solución al problema deben ser estáticos no se permite el uso de Vector ni List ni ninguna colección dinámica. La cantidad máxima de películas ingresadas en la aplicación será de 20.

El código implementado debe corresponder con el diagrama de clase entregado y con el funcionamiento descrito en el documento.