

Proyecto de programación

Fecha de entrega: 16 de setiembre

Objetivo

El objetivo del proyecto es aplicar los conceptos teóricos de la programación orientada a objetos (POO) estudiados en clase, a través del diseño e implementación de una solución eficiente usando POO. Es importante que al desarrollar el proyecto se escriba código correctamente estructurado, debidamente encapsulado y fácilmente reutilizable.

Descripción del proyecto

Se construirá un sistema sencillo de venta de tickets para una empresa de autobuses. El sistema contempla la administración de autobuses(unidades), viajes (carreras) y venta de tickets.

Los archivos del proyecto tienen que entregarse en el formato adecuado para su revisión. No se recibirán proyectos en un formato diferente al indicado. El diagrama de clases se elaborará usando alguna aplicación para ese efecto y se adjuntarán en formato PDF. Durante las lecciones el estudiante recibirá por parte del profesor realimentación para refinar la elaboración del diagrama.

Consideraciones de implementación

El código del proyecto debe estar estructurado adecuadamente, en particular:

- Las clases deben separar correctamente la declaración de la interfaz y su implementación, usando archivos de cabecera (archivos .h) y código fuente (archivos .cpp) por aparte. Los archivos de cabecera deberán siempre usar guardas (`#pragma once` o `#ifndef`).
- Las clases de entidad no deberán contener código de entrada/salida (como salida a la consola usando `cout`, por ejemplo). Sin embargo, se pueden mostrar mensajes de comprobación al efectuar las pruebas del programa, para corroborar que se están ejecutando las funciones de manera correcta.
- El manejo de la interfaz con el usuario debe hacerse por medio de una o varias clases diseñadas para tal efecto (haga uso de la clase interfaz y control).
- Cuando se necesite manejar conjuntos de datos, se definirán e implementarán colecciones. Es importante que el diseño general del programa considere la reutilización de código.
- No se podrán utilizar las colecciones provistas por la STL (*Standard Template Library*) de C++.

Requerimientos del proyecto

Una empresa de autobuses desea un sistema que permita gestionar la venta de los tickets del mismo día, por lo cual la empresa lo busca a usted para que le desarrolle una solución eficiente y funcional haciendo uso del POO.

El sistema debe permitir registrar el nombre y el número de teléfono de la empresa.

La idea es que el cliente pueda llegar a la ventanilla de la terminal a comprar sus tickets con cierta anticipación.

Al vender los tickets no es de interés para la empresa registrar los datos de cada cliente.

La empresa tiene dos terminales y los tickets se venden tanto en la terminal **A** como en la terminal **B**, por tanto al vender un ticket, se debe especificar si se desea el ticket para viajar de **A a B** o de **B a A**. En ambas terminales se realizan varios viajes durante el día, pudiendo salir varias unidades en horas seguidas, todo depende de la demanda y la disponibilidad de unidades en la terminal.

La empresa tiene una flotilla de 5 buses (unidades), para cubrir los diferentes viajes o carreras, cada **unidad** se registra:

- Un número único que la identifica (puede ser un número del 1 al 5)
- Un número de placa
- Una cantidad de asientos
- Una ubicación:
 - (R), significa que en este momento la unidad esta en ruta, es decir que está haciendo una carrera.
 - (A o B), significa en este momento la unidad se encuentra en alguna de las dos terminales
 - (M) significa que en este momento la unidad esta en mantenimiento

Cuando una unidad sale a realizar una carrera debe actualizarse su ubicación en R. Cuando la unidad llega a una terminal debe cambiarse su ubicación a A o B dependiendo de la terminal a la que haya llegado.

Cada unidad de la empresa tiene 96 asientos, distribuidos en dos pisos, cada piso dispone de 48 asientos. Los asientos se identifican por una numeración la cual posee una letra y número. Defina usted mismo la distribución de esta numeración para cada asiento, podría colocar la letra V cuando se trate de asientos junto a la ventana y P cuando sean asientos junto al pasillo, por ejemplo:

Planta Baja

| | | | |
|-----|-----|-----|-----|
| V1 | P2 | P3 | V4 |
| V5 | P6 | P7 | V8 |
| V9 | P10 | P11 | V12 |
| V13 | P14 | P15 | V16 |
| V17 | P18 | P19 | V20 |
| V21 | P22 | P23 | V24 |
| V25 | P26 | P27 | V28 |
| V29 | P30 | P31 | V32 |
| V33 | P34 | P35 | V36 |
| V37 | P38 | P39 | V40 |
| V41 | P42 | P43 | V44 |
| V45 | P46 | P47 | V48 |

Planta Alta

| | | | |
|-----|-----|-----|-----|
| V49 | P50 | P51 | V52 |
| V53 | P54 | P55 | V56 |
| V57 | P58 | P59 | V60 |
| V61 | P62 | P63 | V64 |
| V65 | P66 | P67 | V68 |
| V69 | P70 | P71 | V72 |
| V73 | P74 | P75 | V76 |
| V77 | P78 | P79 | V80 |
| V81 | P82 | P83 | V84 |
| V85 | P86 | P87 | V88 |
| V89 | P90 | P91 | V92 |
| V93 | P94 | P95 | V96 |

A cada **carrera** se le asigna:

- Numero de carrera
- Una única unidad,
- Una hora de salida,
- Una ruta (A-B o B-A),
- Un precio.

La creación de la carrera debe ser realizada con anterioridad a la venta de tickets. Se pueden crear varias carreras al mismo tiempo, solo debe tener cuidado de que una misma unidad no sea asignada a más de una carrera a la vez. Cada unidad únicamente hace un solo viaje al día, esto debido a lo extenso del viaje.

Por carrera debe poderse registrar el monto recolectado.

El **ticket** de venta que se le entrega al cliente, debe indicar:

- El número de ticket (es un consecutivo único),
- Número de asiento,
- El número de la unidad,
- La ruta (A-B o B-A),
- Hora de salida
- Precio.

Cada ticket es para un único asiento. Si un cliente compra un ticket y quiere cambiar de asiento, el sistema debe permitirselo (siempre y cuando haya disponibilidad).

Si un asiento ya está ocupado, obviamente no se puede vender. Es importante visualizar en pantalla el estado de los asientos del bus, para que así el cliente pueda escoger el de su preferencia, es decir se debe poder visualizar los asientos ocupados y disponibles de cada carrera. El cliente puede escoger entre cualquiera de las carreras creadas para el día.

Otras indicaciones

Queda a criterio de cada profesor solicitar defensa del proyecto dentro o fuera de horario de clase.

El proyecto se realizará en grupos de dos estudiantes. El proyecto se entregará por medio del aula virtual, en el espacio designado para ello. No se recibirán documentos ni materiales de otro modo, salvo por indicación manifiesta del profesor. Queda a criterio del profesor utilizar un medio de entrega diferente cuando lo considere conveniente. No se admitirá la entrega tardía del proyecto. El profesor a cargo adjuntará un documento con los criterios de evaluación.

Recuerde incluir un bloc de notas con el nombre completo de cada integrante. Solo uno de los integrantes debe subir el proyecto.

Elabore y entregue el diseño UML en formato pdf, con sus respectivos atributos, métodos y relaciones.

Entregue el proyecto completo en visual 2015.

Distribución de puntos:

En todos los siguientes rubros se evalúa: funcionamiento, eficiencia, diseño, uso correcto del paradigma, respeto y establecimiento adecuado de relaciones.

| Rubros | Puntaje |
|--|------------|
| Registro, consulta y modificación de unidades (listando las unidades por terminal y mostrando información de cada una) | 10pts |
| Inclusión, modificación y consulta de carreras (mostrando toda la información requerida para las carreras, incluyendo el detalle de los asientos y monto recolectado por carrera) | 30pts |
| Venta de tiquetes (incluyendo visualización de asientos con sus respectivos estados, visualización del tiquete recién creado, lista de tiquetes vendidos y la posibilidad de hacer cambios) | 30pts |
| Uso correcto de la clase interfaz y control (se evalúa demás el uso de una interfaz agradable y clara para el usuario) | 15pts |
| Uso adecuado de memoria dinámica en instancia y colecciones (toda la memoria utilizada debe ser dinámica, haciendo la adecuada liberación de la misma). | 10pts |
| UML completo | 5pts |
| TOTAL | 100 |

Cree un menú adecuado que permita visualizar toda la información requerida con el sistema.