APTC101
Semana 3
Sumativa 1: Guía de aprendizaje 1

DESCRIPCIÓN DE LA ENTREGA*:*

Los alumnos resolverán individualmente el caso que se describe más adelante en este documento, que consiste en aplicar los conceptos fundamentales de la POO revisados en la presente unidad. Debe desarrollar el diseño y construcción de la solución elaborando lo siguiente:

- Diseño de la solución del caso a través de un diagrama de clases. Este diagrama debe representar la abstracción de clases, atributos, métodos y relaciones de asociación, agregación y composición según corresponda.
- Programar la solución diseñada usando el lenguaje de programación Java. Debe probar su programación siguiendo las indicaciones del enunciado.

Al término del desarrollo del desafío, antes que se cumpla el plazo de entrega, debe subir a la plataforma habilitada para tal efecto, una carpeta comprimida (con su nombre) con los siguientes productos:

- **Un informe en formato pdf** con al menos los siguientes ítems:
  - Portada.
  - Índice de contenidos.
  - Introducción: Una página donde sintetice la motivación y/o contexto, objetivos del proyecto, propósito del informe y estructura del informe.
  - Análisis y diseño de la solución: debe mostrar el diagrama de clases acompañado de la justificación (o fundamentos) del diagrama. También debe justificar las cardinalidades y los tipos de relaciones que ha decido utilizar. Recuerde usar las buenas prácticas y convenciones de la POO.
  - Conclusiones: debe concluir en términos de los resultados y dificultades de su trabajo con capacidad crítica. Indicar los niveles de logros de sus objetivos y marco de acción futuro de su trabajo.

- ***Carpeta con los programas del proyecto Java***. Incluya en la carpeta un readme.txt que explique las consideraciones para tomar en cuenta al ejecutar su proyecto.

OTRAS CONSIDERACIONES*:*

- El docente responderá las preguntas de los estudiantes y efectuará el monitoreo del avance en el foro dispuesto para este efecto (en la etapa Ponte a prueba), realizando preguntas sobre los conceptos claves para solucionar los requerimientos de información planteados. Además, incentivará la participación en la búsqueda de buenas soluciones.

- El estudiante debe utilizar una herramienta que permita diseñar diagramas de clases con la notación entregada en este curso. No está permitido utilizar notaciones que no se hayan entregado en los apuntes y bibliografía del curso.

- El estudiante puede utilizar cualquier ambiente de desarrollo que le permita programar en Lenguaje Java. En el readme.txt, de la carpeta donde estarán sus programas, indique el **IDE** y las consideraciones que debe tener el evaluador para ejecutar su programa.

- El docente evaluará según los aspectos que se indican en la rúbrica que puede descargar de la plataforma.

ENUNCIADO DEL CASO

La empresa de arriendo de vehículos **"Car-REnt"** ha decidido implementar un sistema informático que permita mejorar la gestión de su empresa. Por lo tanto, el gerente lo ha contratado a usted para diseñar y programar una solución que administre y almacene la información asociada a sus automóviles, clientes, arriendos y devoluciones. El requisito fundamental es utilizar la programación orientada a objetos y sus buenas prácticas. La especificación de lo que el gerente requiere se detalla a continuación:

- Un vehículo se caracteriza por su patente *(largo 8)*, marca, modelo, *año* fabricación (cualquier año desde el 2000 y año actual) y la condición del vehículo. Esta condición puede tomar uno de los siguientes datos: **D, A** o **M**, donde D es condición disponible, A es arrendado y M en mantención. Cuando un nuevo vehículo se ingrese al sistema, inicialmente toma la condición D (disponible) y el sistema además debe preocuparse que los valores de la patente, el modelo y marca se reciban con mayúscula.
- Un cliente se caracteriza por su cédula (largo 10 con penúltimo dígito un guion y último dígito valor entre 0 al 9 o una k), nombre y si está vigente o no (***true***: es vigente, ***false***: no vigente). Cuando un nuevo cliente se ingresa al sistema, su vigencia inicial es ***true.***
- Un arriendo, se caracteriza por un número de arriendo, *el* vehículo, el cliente que lo arrienda, la fecha de arriendo y el número de días arrendado (mayor que 1 y menor que 10). El monto del arriendo se obtiene a partir del precio que el usuario ingresa en el momento de ingresar el arriendo.

- La devolución se caracteriza porque pertenece a un arriendo específico y su fecha de devolución. Esta fecha no puede ser menor a la fecha del arriendo.
- La funcionalidad requerida por el usuario es la siguiente. Usted debe asignar estos comportamientos a la clase que corresponda:

  a) Deshabilitar cliente: Esta operación la ejecuta el gerente del local cuando el estime conveniente y lo único que debe hacer es dejar no vigente al cliente para que no pueda arrendar.

  b) Asignar vehículo en mantención: Esta operación la ejecuta el gerente del local cuando lo estime conveniente y lo único que debe hacer es dejar la condición del vehículo en mantención. Esta condición no es posible asignarla si el vehículo se encuentra actualmente arrendado.

  c) Evaluar arriendo: operación que se ejecuta antes de guardar el arriendo al sistema y valida que el cliente del arriendo esté vigente y que el vehículo de este mismo arriendo tenga condición D (disponible). La función retorna un true (si está ok) o false (si no es posible arrendar).

  d) Ingresar arriendo: Este método evalúa los datos del arriendo instanciado (ejecutando la función c). Si la evaluación fue exitosa, actualiza automáticamente la condición del auto en A y retorna un true, en caso contrario, no guarda el arriendo retornando un false.

  e) Obtener monto a pagar: Este método calcula y entrega el monto del arriendo multiplicando el número de días por el precio diario. Este precio lo entrega el usuario una vez que se ingresa un arriendo al sistema.

  f) Generar ticket de arriendo: método que muestra la información del arriendo ingresado, respetando el formato del siguiente ejemplo. Para mostrar el monto a pagar se invoca al método definido en e).

T I C K E T ARRIENDO D E V E H Í C U L O

NÚMERO ARRIENDO : 23453
VEHÍCULO : CZSL60-6 Toyota Corolla
PRECIO DIARIO : $45000

FECHA	CLIENTE	DÍAS	A PAGAR
10-01-2021	20023066-3/Juan Pérez	3 días	$135000

FIRMA CLIENTE

g) Ingresar devolución de un vehículo arrendado: este método valida de que todos los datos del vehículo devuelto correspondan a los ingresados en el arriendo respectivo. Si son correctos, esta función dejará el auto devuelto en condición D y retornará un true, en caso contrario, retornará un false.

- Además, se han solicitado otros requerimientos adicionales de programación para cautelar la correcta aplicación de la POO:
  a) Las validaciones de los atributos que se implementan en los mutadores, se deben programar usando métodos públicos en la clase respectiva. Estas validaciones se ejecutan en el momento de actualizar (setear) el objeto, por lo tanto, debe considerar, además, que los datos ingresados por el usuario (en el caso que corresponda) y que son ocupados para instanciar los objetos, podrán ser validados usando estos mismos métodos.
  b) Para que los objetos del sistema entreguen al usuario los diversos mensajes generados por el sistema, programar en cada clase un método que reciba el string con el mensaje y genere la salida. La idea es que los mensajes al usuario se concentren en un único método.
- Para probar el funcionamiento del sistema, programe el método main en una clase de prueba con lo siguiente:
  a) Crear un cliente asignando los datos directamente (no use entrada por teclado) y pruebe que las validaciones implementadas en los mutadores están funcionando. Muestre el estado del cliente recién creado.
  b) Crear un automóvil asignando los datos directamente (no use entrada por teclado) y pruebe que las validaciones implementadas en los mutadores están funcionando. Muestre el estado del vehículo recién creado.
  c) Deshabilite al cliente creado y muestre su estado para validar sus datos.
  d) Asigne en mantención al vehículo creado y muestre su estado para validar sus datos.
  e) Pruebe el ingreso de un arriendo ejecutando la entrada de datos directa (sin teclado) por agregación o composición según corresponda. Use datos y/o objetos distintos a los que creó en a) y b). Ingrese dos arriendos: uno que se ingrese satisfactoriamente y otro que no. Para ambos casos muestre mensajes al usuario según corresponda. Para el ingreso del arriendo satisfactorio genere el ticket de arriendo. Muestre el estado del vehículo arrendado para ver si sus datos están correctos.
  f) Finalmente, ingrese dos devoluciones: una que se ingrese satisfactoriamente y otra no. Muestre mensajes al usuario según corresponda. Muestre el estado del vehículo devuelto para ver si sus datos están correctos.

NOTA IMPORTANTE:

No se ponga a programar inmediatamente, ya que si su diagrama de clases es correcto su programa estará correcto. DISEÑE PRIMERO, DESPUÉS PROGRAME.

Por lo menos se debe tomar uno o dos días en diseñar el programa. Pregunte sus dudas y cuando esté seguro, de que las clases y atributos que ha identificado están correctos, que los compartimientos (métodos) están correctamente asignados a la clase respectiva, y que las relaciones representan lo que expresa en enunciado; solo después de esto, comience a programar.

El costo de modificar un programa mal diseñado es muy alto. **No cometa el error en que incurre la mayoría y del que después se arrepiente.**