

Objetivos de unidad

Unidad 3: Estructuras lineales enlazadas

- Utilizar estructuras enlazadas de objetos para modelar grupos de atributos de longitud indeterminada.
- Escribir los algoritmos necesarios para manipular estructuras lineales que almacenan sus elementos enlazándolos entre ellos.
- Construir interfaces de usuario más complejas, utilizando nuevos distribuidores gráficos y nuevos componentes que van a facilitar el despliegue de información y la interacción con el usuario

Unidad 4: Mecanismos de reutilización y desacoplamiento

- Explicar la importancia de desacoplar las clases que hacen parte de un programa y utilizar interfaces para independizar los contratos funcionales de las implementaciones concretas. Esto con el fin de hacer más flexible y fácil de cambiar el programa que se construya.
- Utilizar las interfaces Collection, List e Iterator de Java que permiten la manipulación abstracta de estructuras contenedoras.
- Explicar la importancia de la herencia como mecanismo de reutilización, con el cual es posible construir nuevas clases a partir de clases ya existentes, las cuales han sido diseñadas con el propósito de facilitar la implementación de una familia de entidades que comparten elementos en común.
- Utilizar la herencia como mecanismo de construcción de aplicaciones en Java y entender el papel que juega la clase Object en dicho lenguaje.
- Entender el uso que le hemos dado a la herencia en niveles anteriores, para construir interfaces de usuario y tipos de excepciones.
- Construir interfaces de usuario que incluyan menús de opciones y gráficas simples en 2 dimensiones.

CONDICIONES DEL EJERCICIO

1. Para este ejercicio usted debe realizar el proceso completo de desarrollo de software. Lo que significa que su carpeta de entrega del ejercicio debe tener adentro 5 subcarpetas así:
 - **Planeación del proyecto:**
 1. Enlace al cronograma de trabajo y los formatos PSP. Previamente se deben haber compartido los archivos, con permiso de hacer comentarios, con el respectivo profesor del grupo. Cada archivo compartido debe tener el nombre que identifique su contenido y el nombre de los responsables, por ejemplo: Formatos PSP – Jhon Doe.
 2. Documento de requerimientos funcionales y no funcionales con formato Cupi2.
 - **Diseño:**
 1. Diagrama de clases del mundo, de la interfaz y de las pruebas.
 2. Diseño de pantallas y Mapa de navegación.
 - **Código (Implementación):** Paquetes Java con la implementación de la solución (incluye el paquete *test* asociado a las pruebas).
 - **Test (Pruebas):**
 1. Diseño de los casos de prueba formato Cupi2 (escenarios y casos de prueba en formato Cupi2)
2. Este ejercicio puede realizarse individual o en parejas y debe tener en cuenta las siguientes condiciones:
 - El nombre del archivo comprimido, correspondiente a su entrega de miniProyecto, debe ir nombrado de acuerdo con las especificaciones del documento *Reglas de juego cursos Algoritmos*. De lo contrario, se le asignará la nota solo al estudiante que subió el ejercicio, y si además es subido por más de un estudiante, se le asignará la nota al que tenga la fecha más temprana pues se entiende que ese es el dueño original del ejercicio. Dentro del proyecto, documentación o implementación, no puede aparecer en ninguna otra parte el nombre de los integrantes (ni siquiera en la carpeta dentro del archivo comprimido), si se hace necesaria la referencia a éstos, se debe utilizar los seudónimos de **IntegranteA** e **IntegranteB**. Si se presentase el caso de encontrar los nombres de los integrantes en algún otro lugar del proyecto (que no es el nombre del archivo comprimido), la calificación de la entrega correspondiente tendrá una penalización de 0.5.
 - Ambos estudiantes son responsables del desarrollo total del ejercicio. Es decir, que cualquier separación interna de responsabilidades entre los miembros del grupo no justifica el desconocimiento del proyecto completamente. Si esto llegase a suceder entonces el ejercicio será invalidado para los dos estudiantes del grupo.

ENUNCIADO DEL PROBLEMA

El concesionario “Mis carritos” ha escuchado de las excelentes capacidades en programación de los estudiantes de la universidad y por ello lo ha contactado para realizar un demo de lo que sería su sistema principal.

El concesionario requiere un primer módulo para manejo de información personal, de clientes y empleados. De cada uno de ellos almacena el nombre, apellidos, número de identificación, dirección, teléfono, ciudad, país, celular y correo. Para los empleados guarda además el número de cuenta a donde se realizan sus pagos, nombre del banco, tipo de contrato que tiene con la compañía, el cargo que desempeña y la fecha de su contratación. De cada cliente, se debe guardar la relación de los carros comprados a la empresa y la fecha en que se le entregó el vehículo.

El segundo módulo requerido en la aplicación de “Mis carritos” es para la configuración de los pedidos. El concesionario maneja dos líneas de producto: línea tradicional y línea deportiva de lujo. De todos los carros se debe tener un número de referencia (ID alfanumérico), modelo (año de producción), tipo de combustión (gasolina o diesel), cilindraje, tipo de tracción (4x4 o 2x4), si es de transmisión automática o manual, color, tipo de tapicería y si tiene aire acondicionado. Ahora, los carros deportivos de lujo se hacen exclusivamente sobre pedido, debido a su alto costo, en ellos se incluyen la lista de impuestos por importación y la lista de accesorios de lujos. Actualmente el concesionario tiene una lista base de accesorios con la que trabaja, pero si dado el caso un cliente solicita alguno que no se encuentre en la lista, se debe poder adicionar, así que se espera que la aplicación pueda manejar todos los accesorios de *forma genérica*. De los accesorios registrados cada uno se guarda con un código, nombre, descripción, restricciones de instalación (por ejemplo, cierto accesorio no es compatible con el modelo de chasis 0519) y el costo individual.

En el módulo de pedido se espera poder seleccionar todas las características disponibles de acuerdo a la línea escogida. En el contexto de “Mis carritos”, si se encarga un producto de la línea tradicional, es un pedido programado por producción y si se encarga un carro de la línea de lujo, se denomina pedido de cliente final y en ese caso se debe especificar quién es. Una vez se complete el registro de las cualidades deseadas en el automóvil, se genera el pedido.

Finalmente, el tercer módulo será encargado del seguimiento y control de los pedidos. Dentro de la compañía se tienen 4 grandes estaciones que se deberán ver representadas dentro de la aplicación. Cada pedido al ser generado entra en la cola de la primera estación, *Ensamble*, en este paso los empleados revisan el pedido y crean el esqueleto del automóvil, después, se encuentra la estación *Interiores*, encargada de terminar los detalles de la carrocería interna, una vez pasada esta etapa, se entra en *Exteriores*, que le da el color y los detalles configurados para la apariencia del carro. La etapa final es *Calidad*, aquí los técnicos revisan el producto y determinan si cumple con los estándares de la empresa para entregar al cliente.

El módulo de seguimiento y control de pedidos deberá ser una herramienta para almacenar y mostrar los resultados de los trabajadores en cada etapa. Una vez el pedido es generado se le creará una relación con un carro vacío y a medida que avance por las estaciones, están serán las encargadas de confirmar por medio de la aplicación que los requerimientos del pedido ya hacen parte del producto. El módulo debe permitir ver la cola de pedidos en una estación señalada y promover el primero de ellos a la siguiente etapa. Si un pedido ya se encuentra en etapa Calidad y es promovido, sale del proceso de producción y entra en una lista del concesionario llamada: *listos para la venta*.

“Mis carritos” sugiere que se utilice algún tipo de mecanismo para guardar la información de todo el sistema, pues no quieren tener contra tiempos con pedidos. Si en algún momento el equipo donde se ejecuta la aplicación se apaga, deben poder recuperarse los datos.

RESTRICCIONES DEL PROBLEMA

1. Esta unidad evalúa el uso de estructuras enlazadas definidas por el programador. Por lo tanto se espera que usted NO haga uso de las clases en Java que ya tienen implementada esta estructura.
2. Esta unidad introduce elementos gráficos que deben ser incluidos dentro de su solución.

Publicación: 2 de Septiembre de 2013

Entrega 1: 35% - 14 de Septiembre de 2013

Usted debe entregar los siguientes ítems:

- **Planeación del proyecto:**
 1. Cronograma de trabajo y los formatos PSP
Significa la asignación de responsabilidades específicas a cada uno de los integrantes del grupo, detalle de las tareas a realizar y fechas de revisión entre el grupo de estudiantes. Además de los formatos donde se registre su avance real en el desarrollo del proyecto.
 2. Documento de requerimientos funcionales y no funcionales con formato Cupi2.
- **Diseño:**
 1. Diagrama de clases del mundo.
 2. Diagrama de clases de la interfaz gráfica.
 3. Diagrama de clases de las pruebas unitarias.
- **Código (Implementación):**
 1. Desarrollo completo de la Interfaz gráfica.

RUBRICA DEL EJERCICIO (Entrega 1)

Criterio	Puntos
Planeación del proyecto	<i>Subtotal</i> 40
<i>Cronograma de trabajo y Formatos PSP</i>	20
<i>Estándar de la documentación</i>	5
<i>Compleitud de los requerimientos</i>	15
Diseño	<i>Subtotal</i> 40
<i>Diagrama de clases del mundo del problema</i>	20
<i>Diagrama de clases de la interfaz</i>	15
<i>Diseño de pantallas</i>	5
Código (Implementación –Interfaz gráfica)	<i>Subtotal</i> 20
<i>Cumplimiento de los estándares de programación</i>	1
<i>Cumplimiento de la documentación</i>	4
<i>Cumplimiento de las restricciones del enunciado en cuanto a los temas de la unidad</i>	5
<i>Funcionalidad en general</i>	10
TOTAL	100

Entrega 2: 65% - 2 de Octubre de 2013

Usted debe entregar: los mismos ítems de la entrega 1 (si les ha realizado modificaciones o no) más los siguientes:

- **Diseño:**
 2. Casos de Uso y Diagrama de clases de las pruebas.
- **Código (Implementación):**
 1. Paquetes Java con la implementación de la solución (incluye el paquete test asociado a las pruebas).
- **Test (Pruebas):**
 2. Diseño de los casos de prueba formato Cupi2 (escenarios y casos de prueba en formato Cupi2)

RUBRICA DEL EJERCICIO (Entrega 2)

Recuerde que en esta segunda entrega debe enviar su proyecto completo, con todas las partes detalladas a continuación. Si usted no modificó ninguna de las partes enviadas en la primera entrega, de igual modo debe enviarlas con esta segunda parte y serán calificadas nuevamente aunque esta nueva ponderación solamente será tomada en cuenta en la nota de la segunda

entrega. La nota de la Primera Entrega no cambia. Por tanto, si usted actualizó alguna de estas partes y desea que le sean tenidas en cuenta estas mejoras, no olvide incluir los documentos actualizados (no los anteriores).

Criterio	Puntos
Planeación	<i>Subtotal</i> 10
<i>Cronograma de trabajo y Formatos PSP</i>	4
<i>Estándar de la documentación</i>	2
<i>Compleitud de los requerimientos</i>	4
Diseño	<i>Subtotal</i> 25
<i>Diagrama de clases del mundo del problema</i>	10
<i>Diagrama de clases de la interfaz</i>	5
<i>Casos de uso y Diagrama de clases de las pruebas</i>	5
<i>Diseño de pantallas</i>	5
Código (Implementación)	<i>Subtotal</i> 35
<i>Cumplimiento de los estándares de programación</i>	5
<i>Cumplimiento de la documentación</i>	5
<i>Cumplimiento de las restricciones del enunciado en cuanto a los temas de la unidad</i>	10
<i>Funcionalidad en general</i>	15
Test (Pruebas)	<i>Subtotal</i> 30
<i>Estándar de la documentación</i>	5
<i>Calidad de los escenarios y casos de prueba</i>	10
<i>Funcionalidad en general de las pruebas</i>	15
TOTAL	100