Programación 4

EXAMEN - JULIO 2007

Por favor siga las siguientes indicaciones:

- Escriba con lápiz
- Escriba las hojas de un solo lado
- Escriba su nombre y número de documento en todas las hojas que entregue
- Numere las hojas e indique el total de hojas en la primera de ellas
- Recuerde entregar su numero de examen junto al examen

Problema 1 (25 puntos)

a) Mencione los subtipos del tipo `Collection` de OCL y explique en no más de 5 líneas en qué se diferencian.

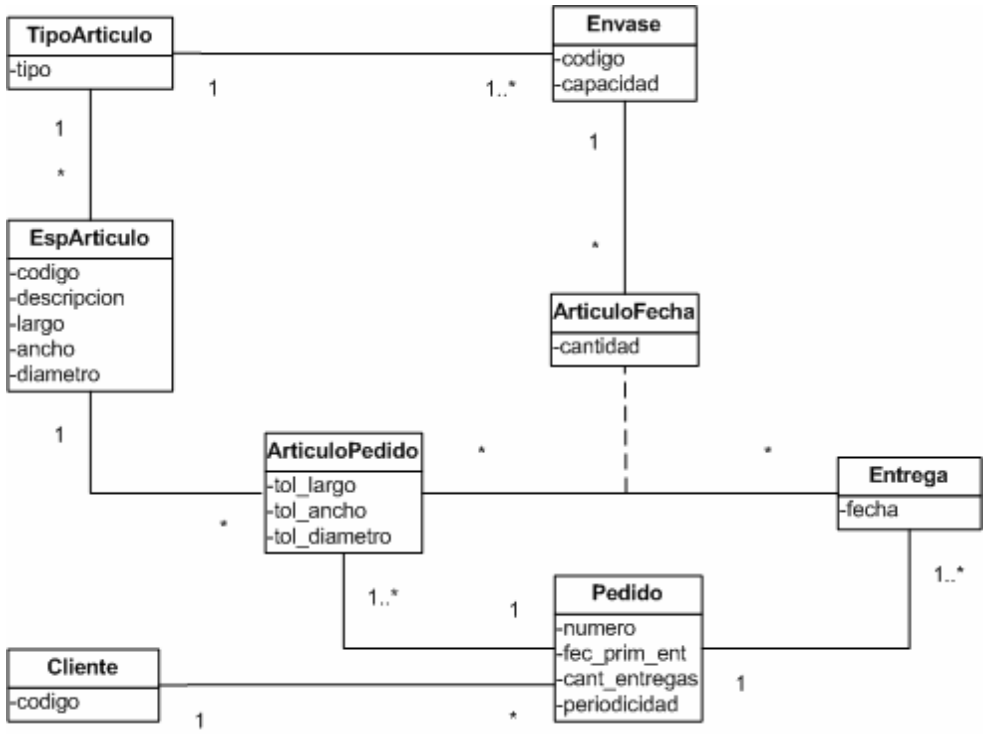
b) Construya un Diagrama de Modelo de Dominio UML para la siguiente realidad. Las restricciones deben ser expresadas en lenguaje natural y en OCL. Modele exclusivamente en base a la información presente en la descripción.

> Una empresa desea mantener información sobre sus proyectos en curso. Cada proyecto tiene un único nombre que lo identifica, la fecha de inicio, la fecha estimada de fin y un empleado en particular que es el líder del proyecto (asignación que puede cambiar con el tiempo).
>
> En un proyecto trabajan uno o más empleados (sin considerar al líder). Un empleado que sea líder no puede trabajar en más de un proyecto, el resto de los empleados sí. Pueden existir empleados que no trabajen en ningún proyecto. Un empleado puede ser líder de a lo sumo un proyecto. Se mantiene registro de cuántas horas un empleado está asignado a proyecto (valor que no cambia mientras dura el proyecto). Un empleado no puede estar asignado más de 10 horas entre todos los proyectos en los que participa.
>
> De cada empleado se mantiene también su supervisor directo. Sólo un empleado líder no es supervisado por alguien más. Un empleado no puede ser supervisor directo de él mismo.
>
> Los proyectos compran insumos a proveedores. De los insumos se conoce un nombre. De los proveedores se conoce el nombre y una dirección. De las compras se desea mantener la fecha en que se realizó, el insumo comprado (solo uno), el proveedor, el proyecto para el cual se hizo la compra, la cantidad del insumo comprado y su costo total. Un mismo proyecto puede comprar el mismo insumo al mismo proveedor solamente en fechas diferentes.
>
> Notas:
> - El modelo NO considera el histórico de proyectos sino solo los que están en curso
> - Para la comparación de fechas en OCL se asume la existencia de un tipo Fecha que soporta los operadores habituales de comparación (=, <, >, <=, >= y <>)

Problema 2 (25 puntos)

a) Definir Contrato de Software e indicar brevemente cuáles son las responsabilidades del proveedor y el consumidor del mismo.

b) Una empresa que se dedica a la producción y venta de artículos de acero y aluminio como ser tubos y piezas para automóviles, desea centralizar los pedidos de producción mediante un planificador de pedidos. Dicho planificador permitirá a los clientes, indicar los artículos que necesita, la forma de envasado de los mismos y las distintas fechas en las que deben ser entregados. La empresa dispone de un conjunto de envases que se encuentran codificados, los cuales son para un tipo de artículo y tienen cierta capacidad definida. A partir de los pedidos realizados por los clientes, la empresa podrá, a partir de su stock, determinar las cantidades a producir, y los plazos que tiene para enviar los productos a sus clientes. Se ha obtenido el siguiente modelo de dominio parcial de la realidad y el caso de uso para el ingreso de un nuevo pedido:



Nombre	Nuevo Pedido	Actores	Usuario
Descripción	Para ingresar un nuevo pedido, el usuario indica el código de cliente, la fecha de la primera entrega, la cantidad de entregas que se deberán realizar y la periodicidad de las mismas (Única Vez, Semanal o Mensual). Con dicha información el sistema determina las fechas de entrega que tendrá el pedido. Luego, el usuario ingresará los artículos a pedir indicando, para cada artículo, el código que lo identifica, las tolerancias que permitirá en diámetro, largo y ancho del mismo y las cantidades a pedir para cada una de las fechas de entrega. El sistema deberá permitir al usuario seleccionar el envase en el que se entregará cada tipo de artículo para cada fecha de entrega. Para ello el usuario tiene que previamente consultar los envases disponibles para un tipo de artículo determinado. En caso de que el usuario no quiera especificar de qué forma se envasarán los artículos, el sistema lo determinará automáticamente. Una vez ingresados todos los artículos, el sistema retorna el identificador del nuevo pedido.		

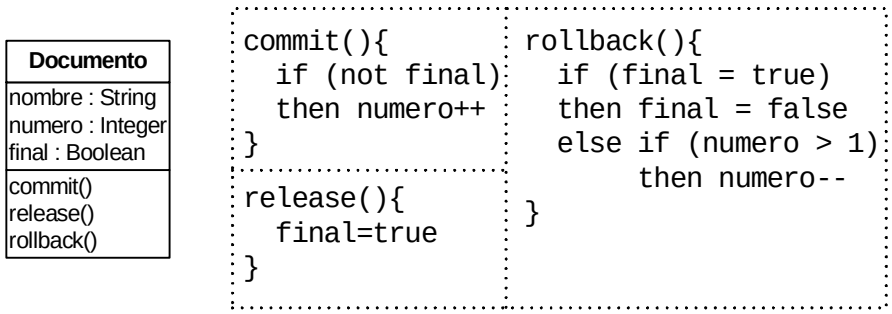
Se pide:

i. Realizar un único Diagrama de Secuencia de Sistema para el caso de uso anterior. Incluya los tipos de los parámetros y valores de retorno de las operaciones.
ii. Indicar las precondiciones y postcondiciones de los contratos de software de las operaciones del sistema identificadas en el DSS de la parte anterior, exceptuando lo que corresponde al envasado automático de los artículos.

Problema 3 (25 puntos)

a) Responda en no más de 3 líneas cada una de las siguientes preguntas
   i. ¿Qué es un controlador?
   ii. ¿Qué tipos de controladores existen y qué operaciones contienen?
   iii. ¿Qué relación existe entre controladores e interfaces del sistema?

b) Un sistema de control de versiones es un sistema que permite administrar las diferentes versiones de un documento facilitando el trabajo concurrente entre varios usuarios. Hemos diseñado un sistema básico que permite conocer la versión de un documento individual sobre la cual trabaja un usuario. El usuario puede realizar cambios sobre un documento y modificar la versión del mismo, así como marcar el documento como final indicando que no se realizarán futuros cambios. Un documento tiene un nombre (que lo identifica), un número de versión (que comienza en 1) y un atributo que indica si el documento está en su versión final. El comportamiento básico del sistema está definido en las siguientes operaciones: `commit()` permite avanzar la versión del documento si el documento no es final, `release()` permite marcar como final la versión de un documento no pudiéndose generar nuevas versiones luego, y `rollback()` permite volver a una versión anterior o marcar el documento como no final. A continuación se presenta el diseño actual del sistema junto con un pseudo-código de su comportamiento.



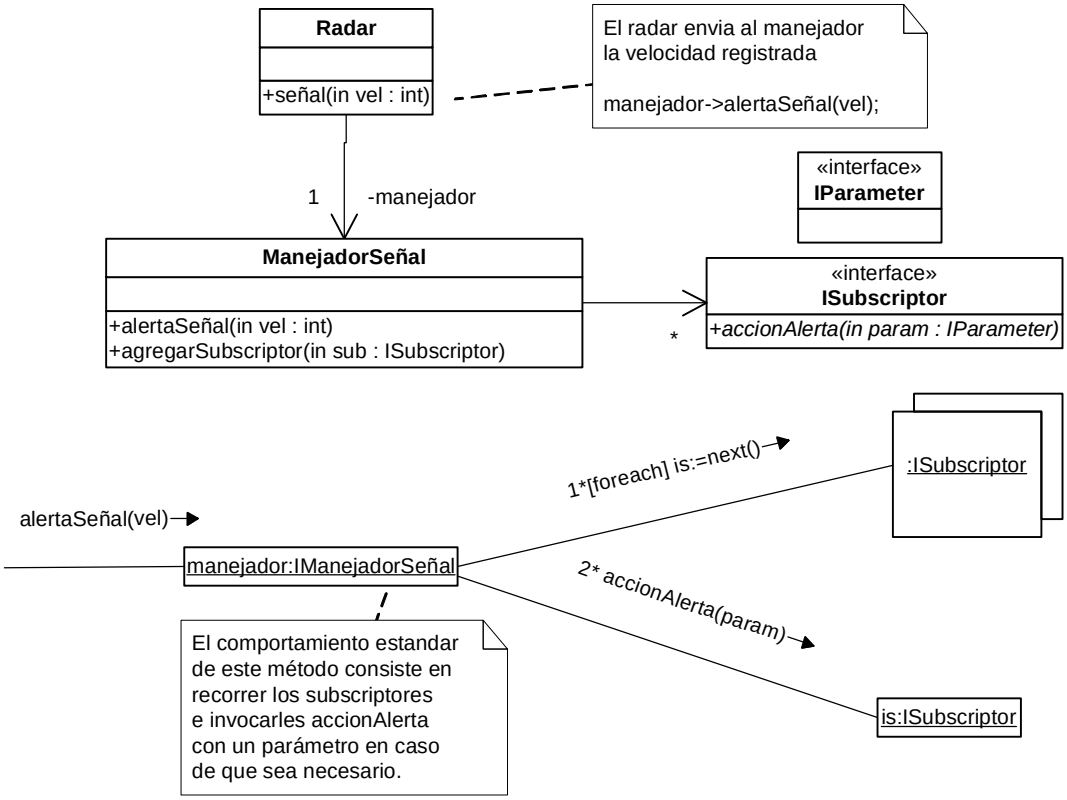
Se desea modificar el diseño de forma tal de permitir que el documento varíe su comportamiento cuando su versión cambie, eliminando el grueso de la lógica condicional.

Se pide:

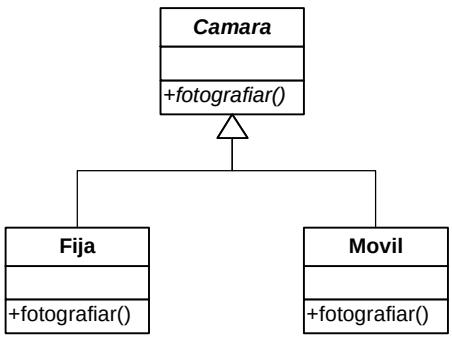
i. Realice el Diagrama de Clases de Diseño de la solución al problema planteado
ii. Realice el Diagrama de Comunicación completo para la operación `rollback()`
iii. Identifique el patrón de diseño utilizado indicando las clases participantes y sus roles

Problema 4 (25 puntos)

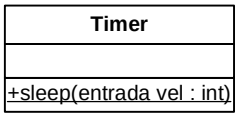
Se está considerando un sistema de control de velocidad de conductores, con el cual se podrá identificar a un infractor mediante fotografías tomadas en el momento de cometer una infracción. Con este fin se adquiere por parte de la empresa un radar que ante una medición de velocidad por encima del límite permitido, dispara una señal que incluye la velocidad registrada, la cual es capturada por una clase `Radar` incluida en la librería del mismo, y ésta finalmente divulga lo ocurrido mediante un manejador a las instancias registradas para este fin. Las instancias a registrar deben implementar la interfaz `ISubscriptor`, esta interfaz tiene un método que recibe como parámetro un objeto de tipo `IParameter` con el cual se permite realizar una acción específica en cada subscriptor. Se muestran a continuación parte de las clases incluidas en la librería y su comportamiento:



La empresa también ha adquirido dos tipos de cámaras fotográficas, unas móviles capaces de seguir un objeto en movimiento y otras fijas. Ambos tipos de cámaras son capaces de accionarse mediante invocaciones a instancias de las siguientes clases, las cuales están dadas y no deben modificarse:



Se desea que las cámaras fijas capturen al conductor cuando éste se acerca a la zona donde están ubicadas las mismas, de manera tal de identificar la matrícula delantera del vehículo y que las móviles se accionen luego para capturar al infractor y en caso de que sea posible la matricula trasera del vehículo. Las cámaras fijas deben tomar las fotografías al instante de registrarse la infracción y las móviles deberán hacerlo luego de las fijas, según un tiempo que dependerá de la velocidad registrada. Para esto se desarrolló una clase `Timer` que posee un método estático que "duerme" la ejecución según una velocidad, la misma se detalla a continuación:



Dado que las cámaras pueden fallar en el disparo o en el seguimiento del objeto en movimiento, se cuenta con varias instancias de cada una de las mismas.

Dadas las compras realizadas se quiere construir el sistema que se propuso, para lo cual se necesita completar el diseño con el fin de utilizar correctamente las librerías provistas.

Se pide: Implemente el `ManejadorSeñal`, las interfaces `IParameter` e `ISubscriptor` y todas las clases que defina para resolver el funcionamiento parcial del sistema descrito.

Notas:

- No es necesario incluir directivas al preprocesador en el código.
- Puede asumir que las clases `Radar`, `Timer`, las cámaras y los siguientes elementos ya están implementados: