Programación Avanzada Examen Julio 2015

Por favor siga las siguientes indicaciones:

- Escriba con lápiz y de un solo lado de las hojas
- Escriba su nombre y número de documento en todas las hojas que entregue
- Numere las hojas e indique el total de hojas que entregue en cada una de ellas
- Comience cada ejercicio en una hoja nueva
- El parcial es individual y sin material.
- Está prohibido el uso de computadoras, tabletas o teléfonos durante el parcial
- Duración: 3 horas.

Ejercicio 1 (35 puntos)

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

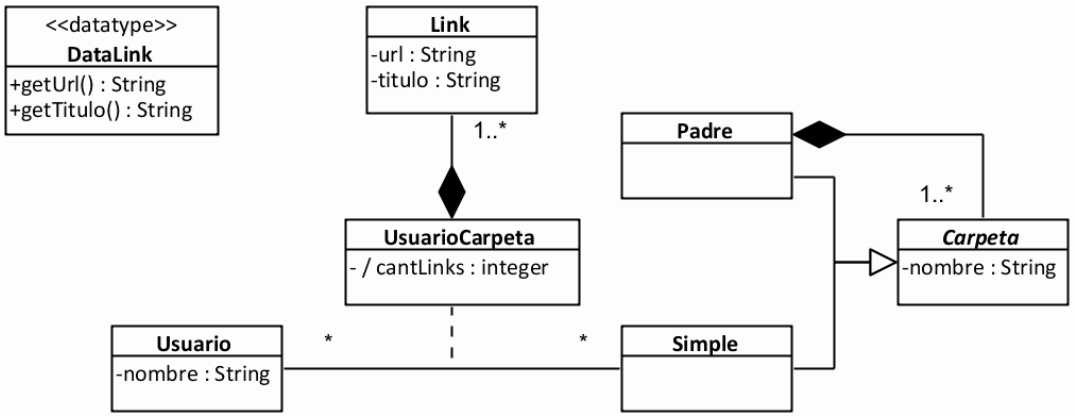
Una Empresa Productora de Televisión desea realizar un software que le permita gestionar varios concursos que la misma produce. Cada uno de estos concursos está formado por un conjunto de parejas que compite en una disciplina determinada (baile, canto, etc.), pudiendo existir varios concursos para una misma disciplina.

Un concurso se identifica por el nombre y tipo de disciplina, es de carácter eliminatorio y se realiza en distintas etapas que se llevan a cabo en una fecha determinada. Cabe notar que no puede haber dos etapas de un mismo concurso con la misma fecha. Como resultado de cada etapa, dos o más parejas quedarán sentenciadas, lo que habilitará a una votación del público. La votación se realiza por vía telefónica, por lo que a cada pareja se le asignará un número telefónico (que la identifica) al comenzar el concurso.
La pareja sentenciada con menor cantidad de llamados será eliminada del concurso.

Cada pareja participante está formada por un soñador y un famoso identificados por su cédula. Otros datos de interés son el nombre y la edad de los participantes. Para los soñadores se necesita saber además su ciudad natal, si es profesional y una descripción del sueño a cumplir en caso de ganar el concurso. Las reglas establecen que a un soñador sólo se le permitirá participar en un concurso mientras que los famosos pueden participar en más de uno. No está permitido que un famoso pueda participar con más de un soñador de un mismo concurso.

Problema 2 (30 puntos)

Se desea construir un sistema para un sitio web de almacenamiento de enlaces donde los usuarios pueden compartir *Links* interesantes a páginas web y almacenarlos en diferentes carpetas que son visibles desde todo el sitio. Las carpetas son compartidas y los usuarios pueden subir links en las carpetas independientemente de que otro usuario haya subido algún link en ella. Las carpetas forman una estructura arborescente de manera similar a un sistema de archivos. Hay dos tipos de carpetas: las *Carpetas Simples* que son las que contienen los Links y las carpetas *Padre* que agregan a un conjunto de carpetas.
De cada *Link* se conoce la URL a la página web (que lo identifica dentro de la carpeta) y su título. De los usuarios se conoce el nombre que los identifica. Es de interés saber para cada usuario cuantos links en total subió a cada carpeta. Los links pueden estar solamente en Carpetas Simples. De una carpeta se conoce su nombre que la identifica.
De acuerdo a los requerimientos el equipo de analistas realizó un modelo de dominio que se deberá respetar durante el diseño.



La intención es que el equipo que usted integra diseñe la siguiente operación del sistema.

getLinks(usr: String, N: integer, carp: String):Set(DataLink)	
Descripción	Obtiene los datos de los links subidos en Carpetas Simples contenidas en la carpeta de nombre carp por el usuario de nombre usr. Sólo se consideran los links subidos en Carpetas Simples donde el usuario haya subido al menos N links. Si la carpeta carp es Simple sólo se busca en esa carpeta. De lo contrario se busca en las **todas las** carpetas hijas de carp.
Precondiciones	Existe una instancia de Usuario con nombre usr Existe una instancia de Carpeta (Simple o Padre) de nombre carp
Postcondiciones	Devuelve un Set con data values de DataLink que correspondan a los Links que están asociados a una instancia de UsuarioCarpeta uc que cumple que • uc está asociada al usuario de nombre usr • uc.cantLinks >= N • uc está asociada con una instancia de carpeta Simple de nombre carp o uc está asociada con una instancia de carpeta Simple que es hija de una carpeta de nombre carp.

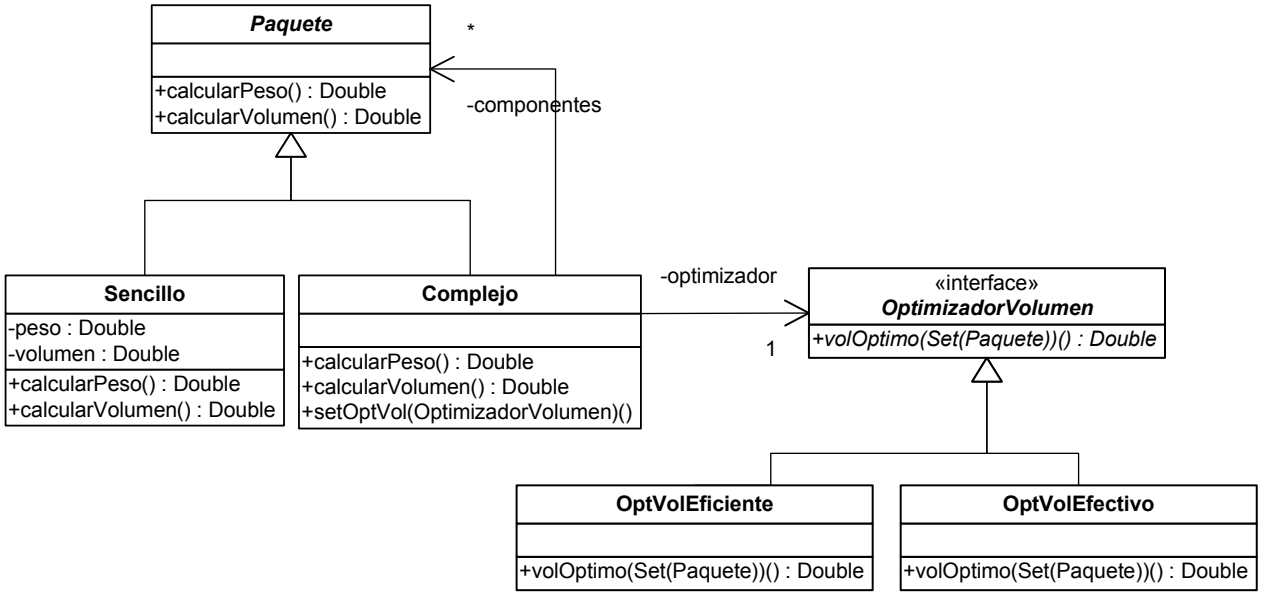
Se pide

1. Realizar el Diagrama de Comunicación de getLinks().
   - Indicar visibilidades en los diagramas
   - Indique claramente el resultado de cada operación utilizando notas.

2. Realizar el Diagrama de Clases de Diseño de la solución

Problema 3 (35 puntos)

Una empresa de servicios de transporte interurbano de objetos frágiles, ha encomendado el desarrollo de un sistema de apoyo a la gestión en el armado de los paquetes chicos y medianos. Los objetos son empaquetados en cajas, ya sea de forma individual o agrupados. Los paquetes así conformados, pueden a su vez ser parte de otros paquetes más grandes. La figura muestra el diseño de la estructura determinada para representar la organización de los paquetes.



Para organizar la carga en los vehículos, es de interés poder estimar el peso y volumen de los paquetes. El cálculo del peso es relativamente sencillo, mientras que el del volumen implica la ejecución de un algoritmo de optimización de ubicación de volúmenes, que tiene un alto costo computacional. Por este motivo, para la estimación del volumen de un paquete que está formado por otros, se utiliza un optimizador de volúmenes. Existen varias versiones de dicho optimizador: algunas estiman de mejor forma el volumen con un alto costo computacional (efectivas), mientras que otras hacen una estimación menos ajustada pero ejecutan de manera más rápida (eficientes).

A su vez, se deben tener en cuenta las siguientes consideraciones:

- El peso y el volumen de un paquete Sencillo se calculan directamente a partir de sus atributos.
- El peso de un paquete Complejo es la suma de los pesos de sus componentes.
- El volumen de un paquete Complejo se calcula invocando a la operación valOptimo de la interfaz Optimizador-Volumen, la cual recibe una colección con los componentes del paquete en cuestión.
- La construcción de un paquete Complejo se realiza a partir de sus componentes y de una instancia de una clase que implemente la interfaz OptimizadorVolumen.
- La destrucción de un paquete Complejo implica la destrucción de todos sus componentes.

Se pide:

Implementar completamente en C++ las clases Paquete, Sencillo, Complejo y la interfaz OptimizadorVolumen.

Observaciones:

- Puede suponer la existencia de la interfaz ICollectible e implementaciones de ICollection (clase List) e IIterator según sea necesario.
- Es posible utilizar las clases set<T> o vector<T> de la STL.
- Las implementaciones deben incluir constructores, destructores y operaciones set y get donde corresponda.
- No incluir directivas al precompilador.