

Programación Orientada a Objetos

Práctica N° 6 – 2019

Para cada una de los siguientes problemas plantee una solución identificando los **objetos que intervienen**, su **estado**, su **comportamiento** y las **relaciones** entre ellos.

1. Implementar una solución orientada a objetos que permita recorrer los árboles implementados en el Práctico 2, en pre-orden. El diseño de la solución propuesta debe permitir recorrer uno a uno los nodos a medida que se piden.

2. Work Breakdown Structure (Wbs)

Una Work Breakdown Structure (Wbs) es una estructura de árbol completa y jerárquica (desde lo más general a lo más específico) de tareas que necesitan ser realizadas para completar un proyecto. Un ejemplo de una estructura Wbs para las actividades de pintar un cuarto sería la siguiente:

- +Preparar Materiales
 - Comprar pintura
 - Comprar Pinceles/Rodillos
 - Comprar Lijas
- +Preparar el cuarto
- +Preparar paredes
 - Remover empapelado
 - Sacar toda la decoración desmontable
 - Sacar apliques de luz
- +Proteger elementos
 - Cubrir el piso con diarios
 - Cubrir enchufes/bordes con cinta de papel
 - Cubrir los muebles con sabanas viejas
- Pintar el cuarto
- +Limpiar el cuarto
 - Guardar la pintura que sobro
 - Limpiar los pinceles y/o rodillos
 - Desechar los diarios
- Remover los cobertores de los muebles

A cada tarea terminal del Wbs se le pueden asignar recursos. Los recursos pueden ser materiales (computadora, mesa, etc) o personas (Juan Perez, Roberto Giord, etc). Estos recursos pueden ser de dos tipos: exclusivos (es decir que pueden estar asignados a una única tarea en un tiempo dado, por ejemplo una persona) o compartidos (recursos que se pueden compartir en tareas, por ejemplo una impresora).

Las tareas terminales poseen una fecha de inicio y una fecha de finalización estimada, así como también fecha de inicio y finalización real. También se guarda si una tarea ya se completó, está en espera o en procesamiento. La fecha de inicio de una tarea compuesta queda determinada por la fecha de inicio más temprana de sus subtareas. De manera análoga, la fecha de finalización de la tarea compuesta queda determinada por fecha de finalización más tardía de sus subtareas.

Programación Orientada a Objetos

Práctica Nº 6 – 2019

Opcionalmente una tarea tiene una lista de tareas predecesoras, es decir, una lista de tareas que deben estar completadas para poder realizar la tarea en cuestión.

Servicios que deben ser provistos por el administrador de Wbs:

- Dada una fecha, devolver la lista de tareas que se pueden realizar (tener en cuenta la lista de tareas predecesoras).
 - Devolver la lista de tareas que están atrasadas (fecha de finalización estimada > fecha actual)
 - Dado un recurso devolver la lista de tareas que lo utilizan.
 - Dada una tarea devolver los recursos involucrados (en caso de no ser terminal es la suma de todos los recursos de las tareas que lo componen).
 - Duración estimada de una tarea.
 - Consultar si es posible asignar el recurso X a la tarea YY (tener en cuenta las fechas de utilización de los recursos y si los mismos son o no exclusivos).
3. Modificar el sistema encargado de la administración del Wbs del ejercicio anterior para que sea capaz de cambiar la forma en la cual almacena los elementos. Inicialmente, existen dos tipos de estructuras que se utilizan para almacenar las tareas, una con un arreglo de 50 elementos y la otra con una lista. El sistema debe proveer mecanismos para cambiar la estructura dinámicamente (tener en cuenta que el sistema debe poder adecuarse a nuevas estructuras fácilmente). El reemplazo debe realizarse en todos los lugares en los que se utilice la estructura.

4. Supermercado

Un supermercado factura las compras que realiza un cliente eventual y para esto emite una factura, que además de calcular el monto total de una compra, detalla para cada producto comprado el total de unidades de ese producto y el valor total de estas unidades (precio * cantidad). Existen además casos excepcionales en los cuales se realiza un descuento: por pago de contado (10%) y por premios que otorga el supermercado (50 % - no tiene descuento por pago contado).

Algunos productos del supermercado están gravados por impuestos (su valor = precio + impuestos provinciales + impuestos nacionales) y otros no. Para el caso de los productos con gravamen se pueden distinguir gravámenes bajos o altos para cuyo cálculo se establece:

Gravámenes Bajos:

- impuestos provinciales: $\text{precio} * 10\% + \$0.10$
- impuestos nacionales: $\text{precio} * 5\% + (\$0.25 \text{ si el año es par})$

Gravámenes Altos:

- impuestos provinciales: $\text{precio} * 15\% + (\$.5 \text{ si es domingo})$
- impuestos nacionales $\text{precio} * 10\% + \$0.50$

Cuando se realiza una compra se debe actualizar el stock de productos disponibles. El stock es también actualizado cuando se recibe una partida de productos que detalla los productos que adquirió el supermercado con su respectiva cantidad. Se debe poder consultar cuáles son los productos que escasean en el stock. Para determinar cuáles son los productos que

Programación Orientada a Objetos

Práctica N° 6 – 2019

escasean el supermercado utiliza algunas de las siguientes políticas (solo una a la vez, y el formato no varía):

- Se llega a un mínimo permitido para ese producto.
- Se llega a un 120% del mínimo permitido para ese producto
- Se llega a un 90% del mínimo permitido para ese producto
- Se llega a un 140% del mínimo permitido para ese producto

Implemente en Java todos los métodos involucrados en responder:

- El total de dinero facturado en una fecha dada
- El total de dinero facturado en un periodo determinado
- El producto que más recaudo en una fecha dada
- El producto que más recaudo en un periodo determinado
- Consultar cuales son los productos que escasean en el stock

5. Servidor de páginas Web

Un servidor de páginas web, además de permitir el almacenamiento y recuperación de páginas, brinda servicios de transformación y adaptación de las mismas para adecuarlas a los estilos que requieren sus clientes. Para esto el servidor permite definir estilos de transformación que luego serán aplicados a las páginas cuando éstas sean solicitadas. Un cliente solicita una página diciendo además qué estilo prefiere.

Una página se compone de texto (múltiples párrafos), imágenes y vínculos a otras páginas, estructurados en cualquier orden según cada página en particular.

Las transformaciones que se pueden aplicar a las páginas son las siguientes:

- Agregar al final de la página el tamaño de la misma en bytes. (cantidad de caracteres párrafo + bytes imágenes + tamaño del documento referenciado por el vínculo)
- Agregar al final de la página la cantidad de componentes (cantidad de párrafos + cantidad de imágenes + cantidad de vínculos)
- Agregar fecha actual
- Agregar un encabezado al comienzo de la página
- Agregar un pie de página
- Reducir el tamaño del documento (reducir imágenes y tamaño de la fuente en cierta escala - cada párrafo puede definir su propio tamaño de fuente)
- Aumentar el tamaño del documento (aumentar imágenes y tamaño de la fuente en cierta escala - cada párrafo puede definir su propio tamaño de fuente)
- Un estilo define que transformaciones aplicar y en que orden, por ejemplo:
 - estilo "defecto" aplica transformación 1,3 y 6.
 - estilo "grande" aplica 3 y 7, etc...

Adicionalmente debe poder agregarse nuevas transformaciones de manera sencilla.

También se debe tener en cuenta que **no** se puede modificar la página original sino se debe devolver una copia de la página transformada.

Nota: el agregado de elementos (pie de página, fecha, etc.) se realiza mediante el agregado de párrafos a la página en la posición correspondiente.

Programación Orientada a Objetos

Práctica N° 6 – 2019

6. Catalogador de recursos multimedia

Un Catalogador de recursos multimedia permite mantener información de programas de software, música, películas y libros digitales. Cada recurso posee un nombre, una fecha de adquisición y una ubicación en las estanterías (número de biblioteca, número de estante y posición en el estante). Los recursos pueden estar almacenados en CDs o en DVDs. Adicionalmente se permite que el usuario agregue nuevos atributos a cada recurso.

También ciertos atributos se derivan en base a otros atributos del recurso. Por ejemplo a continuación se muestra como se podría derivar ciertos atributos (**las formas** de calcular los atributos derivados son solo las que se detallan aquí):

- Atributo descripción: **concatenación** atributo nombre + atributo utilidad
- Atributo horas préstamo: atributo cantidad prestamos * atributo promedio horas préstamo
- Atributo utilización: **si** atributo retiro > 1000 **entonces** "alta" **sino** "baja"
- Atributo utilización2: **si** atributo retiro > 1000 **entonces** atributo cantidad prestamos * atributo promedio horas préstamo sino 0

(Todos los anteriores son ejemplos de atributos que agregó el usuario)

Adicionalmente, se desea poder armar listados ordenados (ascendente o descendente) por múltiples atributos de los recursos. En caso de igualdad de dos recursos para un atributo dado se debe permitir ordenarlos por los siguientes criterios de orden.

Ejemplo: primero por número de biblioteca (ascendente), luego por atributo nombre (descendente) *(se ordenan en forma ascendente por el valor del atributo número de biblioteca, para los elementos con el mismo valor se ordenan luego por el atributo nombre de manera descendente.)*