

Exámenes teóricos y prácticos de diseño de software

28 de septiembre de 2019

Índice

1. Ejercicios teóricos	3
2. Ejercicios prácticos	3
2.1. Empresa de colectivos	3
2.1.1. Requerimientos	3
2.1.2. Módulos 2MIL	4
2.1.3. Guía de módulos	4
2.1.4. Estructura de objetos	4
2.2. Control de stock	4
2.2.1. Requerimientos	4
2.3. Control de archivos y directorios	5
2.3.1. Requerimientos	5
2.3.2. Módulos 2MIL	6
2.3.3. Guía de módulos	6
2.4. Estación de peaje	6
2.4.1. Requerimientos	6
2.4.2. Módulos 2MIL	6
2.4.3. Guía de módulos	6
2.4.4. Estructura de uso	6
2.5. Administrador de errores	6
2.5.1. Requerimientos	6
2.6. Administrador de eventos	7
2.6.1. Requerimientos	7

2.6.2.	Módulos 2MIL	8
2.6.3.	Guía de módulos	8
2.6.4.	Estructura de uso	8
2.6.5.	Estructura de herencia	8
2.7.	Inmobiliaria	8
2.7.1.	Requerimientos	8
2.7.2.	Módulos 2MIL	9
2.7.3.	Guía de módulos	9
2.8.	Control de acceso a edificio	9
2.8.1.	Requerimientos	9
2.8.2.	Análisis de cambio	9
2.8.3.	Módulos 2MIL	9
2.8.4.	Guía de módulos	9
2.8.5.	Estructura de uso	9

1. Ejercicios teóricos

1. Enuncie el criterio sugerido por Parnas para que $A \text{ REC } B$.
2. Suponga que en un cierto diseño existe un módulo M , con algunos procedimientos en su interfaz. Digamos que es necesario extender la funcionalidad de algunos de esos procedimientos sin perder el módulo original y sin tener que duplicar código. Además, la aplicación debe interactuar con las instancias de M y con las del que tiene funcionalidad extendida de la misma forma. Muestre como sería su diseño para este caso.
3. Enuncia la definición de herencia que se comentó en clase y enumere sus ventajas y desventajas.
4. Explique el método propuesto por Parnas para diseñar un sistema de software.
5. Defina módulo, interfaz e implementación.
6. Defina el concepto de herencia y extienda la metodología de Parnas para que lo tenga en cuenta.
7. Indique el criterio de corrección para un diseño. Justifique su respuesta.

2. Ejercicios prácticos

2.1. Empresa de colectivos

2.1.1. Requerimientos

Una empresa de colectivos de larga distancia desea un sistema para manejar la reserva y venta de pasajes. Un pasajero le indica al empleado de la ventanilla fecha de viaje, origen y destino; el empleado verifica que haya lugares disponibles y luego el pasajero le indica el asiento que desea (o uno que este libre). Claramente no se puede vender dos veces el mismo asiento en el mismo colectivo. Tener en cuenta que los viajes muchas veces tienen paradas intermedias y que hay pasajeros que compran tramos intermedios. Por lo tanto, se debe tratar de maximizar el uso del colectivo sin mover a la gente de asiento durante todo el viaje que haya comprado. Cada colectivo

asignado a un viaje tiene una configuración de asientos diferentes, aunque existen modelos de colectivos tales que todos tienen la misma configuración. No siempre el viaje entre A y B de los lunes a las 8:00 tiene asignado el mismo colectivo.

2.1.2. Módulos 2MIL

2.1.3. Guía de módulos

2.1.4. Estructura de objetos

2.2. Control de stock

2.2.1. Requerimientos

Se trata de los requerimientos para un sistema de control de stock que será vendido a empresas de distintos rubros (y posiblemente de distintos países).

El stock se compone de categorías, subcategorías, sub-subcategorías de productos y de otras agrupaciones de productos que pueden anidarse indefinidamente. Cualquiera de estas divisiones tiene un nombre y una descripción. Los productos tienen nombre, descripción, proveedor, cantidad, etc.

El sistema debe impedir que el stock de cada producto no cumpla ciertas condiciones respecto de la cantidad de ese producto u otros. Desconocemos de ante mano las diversas estrategias de control de stock que tendrán nuestros potenciales clientes. Por el momento debemos implementar una estrategia muy simple que comunique al usuario cada vez que la cantidad de un producto cae por debajo de cierto número crítico (obviamente este número no es el mismo para todos los productos). También desconocemos la forma en que el usuario querrá que estos eventos le sean comunicados (en la pantalla de su PC, mediante un mail, o un SMS, etc.).

Las dos operaciones básicas sobre el stock son ingreso de nueva mercadería y salida de mercadería existente pero también son importantes las operaciones para dar de alta o baja una categoría, un producto o reacomodar un producto en otra categoría. Creemos que el ingreso o egreso de mercadería podrá detectarse, tal vez, por lectores de código de barras, carga manual o algún otro mecanismo similar. Además el usuario debe poder consultar el stock de diversas formas y desde estaciones de trabajo muy diferentes (por ejemplo una PC, una laptop, una PDA, etc.); mas aun se espera poder hacer análisis estadísticos sobre la evolución del stock (como el sistema recién

esta siendo concebido no sabemos con precisión cuales son estos análisis, esperamos tener feedback de nuestros clientes para ir implementando tales funcionalidades). Obviamente los datos del stock se guardaran en una base de datos que suponemos el cliente nos impondrá (o mejor, queremos poder adaptarnos al DBMS o repositorio que el cliente tenga).

2.3. Control de archivos y directorios

2.3.1. Requerimientos

Se desarrollara un sistema operativo que impondrá diversas políticas de control de acceso sobre los archivos y directorios. El control de acceso podrá realizarse sobre la base de datos tales como el identificador de usuario, el identificador de proceso, los grupos a los cuales pertenezca el usuario, los archivos que ya tenga abierto el usuario o el proceso, el programa que esté solicitando acceso, los atributos de acceso del archivo y/o la operación de acceso que se solicite (lectura, escritura, modificación de atributos de seguridad). Además cada vez que un proceso solicite acceso a un archivo, el sistema operativo deberá avisar tal suceso a un componente que gestiona ese tipo de información. Este componente guardara los datos de los sucesos que se le informen según una política que se establecerá en base a los datos mencionados mas arriba.

Ítems de cambio

- Se desconoce cuales son todas las políticas de control de acceso.
- Deberá ser posible combinar políticas de control de acceso de manera tal que el acceso se permita o se niegue según el resultado que arroje cada una de ellas.
- Podría ser necesario que alguna política utilice datos que inicialmente no fueron considerados.

2.3.2. Módulos 2MIL

2.3.3. Guía de módulos

2.4. Estación de peaje

2.4.1. Requerimientos

Considere el problema de la estación de peaje visto en clase. Se había tenido en cuenta como cambio probable la incorporación de nuevos medios de pago. Medios de pago como tarjetas de crédito o cheques contienen información que debe estar impresa en el ticket que se le entregara al conductor. Estos datos varían considerablemente de medio de pago en medio de pago. Además, en el diseño presentado en clase, el módulo Control debía acceder a la función `debePagar()` del módulo `TablaValoresVehiculos` lo cual no es muy prolijo debido a que un módulo de alto nivel usa un módulo de bajo nivel.

Otro problema con el diseño original es que cada vez que se agrega un nuevo medio de pago hay que modificar el módulo `RecepcionPago` para que este consulte al nuevo medio de pago. Un detalle menor es que el módulo Control quedaba bloqueado hasta que la función `pago()` del módulo `RecepcionPago` retorna.

2.4.2. Módulos 2MIL

2.4.3. Guía de módulos

2.4.4. Estructura de uso

2.5. Administrador de errores

2.5.1. Requerimientos

Los componentes de un sistema deben comunicarse entre sí. Un componente anuncia un error y otro componente analiza el error y eventualmente lo comunica al usuario. No todos los componentes anuncian errores, ni todos los componentes analizan errores, ni todos los componentes comunican los errores al usuario. En otras palabras, puede ocurrir que uno de los componentes que comunican errores al usuario, sea invocado recién después de que se hayan dado varios errores los cuales deben preservarse hasta ese momento.

Por otro lado, hay errores que tienen parámetros y los hay que no los tienen. Por ejemplo: «El archivo *carta.doc* no existe en el directorios *Cartas*»

es un error que tiene dos parámetros; mientras que «Comando inexistente» es un error que no tiene parámetros.

1. Usted debe diseñar y documentar adecuadamente todos los módulos que sean necesarios para proveer un sistema de errores versátil y suficientemente general; en particular los mensajes que se comunican al usuario se deben poder modificar en tiempo de ejecución.
2. Además debe explicar (sugerimos hacerlo mediante un ejemplo) como los módulos del sistema comunican los errores, los analizan y los comunican al usuario.

2.6. Administrador de eventos

2.6.1. Requerimientos

Se espera que los componentes de un sistema puedan comunicarse entre sí pero sin que se conozcan entre sí. En otras palabras, un componente del sistema no sabe de la existencia de otros componentes del sistema, por lo tanto se recurre al concepto de evento. Los componentes anuncian o publican eventos (sin saber quien los esta esperando) y a su vez se anotan o *suscriben* a eventos (que tal vez nunca aparezcan).

Un componente especial, llamado *administrador de eventos* es el encargado de intermediar esta forma de comunicación. Cada suscriptor manifiesta interés en un evento suscribiendo una subrutina en su interfaz a ese evento. La suscripción se hace ante el administrador de eventos. Los componentes publican eventos llamando al administrador de eventos. La tarea del administrador de eventos es, entonces, invocar a cada una de las subrutinas suscritas a un evento cada vez que ese es publicado (por cualquier componente del sistema).

Cada evento tiene un nombre y una serie de parámetros fija por tipo de evento. El administrador de eventos pasa los parámetros del evento al suscriptor.

Ítem de cambio

- La política de invocación del administrador de eventos.
- Debería ser posible agregar nuevos tipos de eventos sin tener que modificar el administrador de eventos.

- Debería ser posible alterar la lista de parámetros de un evento sin tener que modificar el administrador de eventos.

2.6.2. Módulos 2MIL

2.6.3. Guía de módulos

2.6.4. Estructura de uso

2.6.5. Estructura de herencia

2.7. Inmobiliaria

2.7.1. Requerimientos

Una inmobiliaria desea publicitar las propiedades que vende, compra o alquila a través de la Web. Los clientes que desean comprar o alquilar deberán poder buscar de diversas formas entre las propiedades disponibles. Una vez encontrada la propiedad podrán ver sus detalles e incluso fotos o videos. Si una o mas propiedades les interesan podrán concretar una cita, siempre a través de la aplicación Web, para poder visitarla.

Los clientes que deseen que la inmobiliaria les venda o encuentre inquilinos para sus propiedades podrán completar un formulario on-line con todos los datos de aquellas. El sistema deberá avisar por correo electrónico a los empleados de la firma que hay disponible una nueva propiedad y estos determinarán si la aceptan o no. Cualquiera sea la decisión el sistema se la comunicara al propietario. Si el empleado acepta la propiedad esta pasa al conjunto de propiedades disponibles.

Cuando una propiedad ha sido vendida o alquilada se la retira del conjunto de propiedades disponibles.

Ítem de cambio

- Las opciones de búsqueda podrían ampliarse en el futuro.
- En lugar de que la aplicación la utilicen personas podría ser utilizada por sistemas similares de otras inmobiliarias.
- Los empleados de la firma que reciben los avisos de las nuevas propiedades podrían ser otros.

- Respecto del ítem anterior, el medio (correo electrónico) por el cual reciben la noticia podría cambiar.
- Podría requerirse que el grupo de empleados que recibe los avisos de nuevas propiedades disponibles sea diferente del grupo que puede aceptar o rechazar una propiedad.

2.7.2. Módulos 2MIL

2.7.3. Guía de módulos

2.8. Control de acceso a edificio

2.8.1. Requerimientos

Un edificio cuenta con un mecanismo automático de control de acceso. Quien quiera entrar caminando por la puerta o con su auto por el portón debe deslizar una tarjeta magnética por el lector correspondiente. El sistema debe verificar que es una tarjeta válida y, utilizando una cámara fotográfica digital móvil, obtener una imagen del auto o de la persona según sea. La cámara puede moverse hacia la izquierda, derecha, arriba y abajo. La imagen debe enviarse a una base de datos junto con la fecha, hora y número de la tarjeta. Una vez obtenida la imagen el sistema debe abrir la puerta o el portón y cerrarlos después de cierto tiempo. En algunas circunstancias puede requerirse a la persona que teclee un PIN en un teclado ubicado junto a la lectora de tarjetas.

2.8.2. Análisis de cambio

2.8.3. Módulos 2MIL

2.8.4. Guía de módulos

2.8.5. Estructura de uso