

1. Vocabulario y conceptos

1. Enuncie y explique la sugerencia de Parnas para determinar si *A REC B* o no.

Solución COMPLETAR.

2. Se ha dicho varias veces en clase que métodos como *modificar*, cuya implementación es realizable con métodos como *alta* y *baja*, no deben incluirse en la interfaz del mismo módulo.
 - a) Indique dónde ubicaría estos métodos (*modificar*).
 - b) Analice la ventaja o desventaja de no incluirlos en el módulo con respecto a:
 - 1) Cambios en la representación de los datos.
 - 2) Cambios en los algoritmos de procesamiento de *alta* y *baja*.
 - 3) Cambio en la interfaz de *alta* o *baja*.
 - c) Explique las ventajas y desventajas de la herencia en el DOO.

Soluciones

- a) COMPLETAR.
 - b)
 - 1) COMPLETAR.
 - 2) COMPLETAR.
 - 3) COMPLETAR.
 - c) COMPLETAR.
3. Enuncie las diferencias entre los tres tipos de diseño vistos en clase.

Solución COMPLETAR.

4. Explique las razones por las cuales un método en la interfaz de un módulo que sólo debe modificar el estado de este, no debería indicar si fue posible o no modificar el estado. Explique la solución correcta a este problema.

Solución COMPLETAR.

5. Efectúe un análisis comparativo entre los estilos de diseño funcional y basado en la ocultación de información.

Solución COMPLETAR.

6. Un sistema mide cierta magnitud a través de un sensor y según ciertas reglas debe controlar el movimiento de una máquina externa. Indique cuál es el mejor de los siguientes diseños justificando su respuesta:
 - a) El módulo *A* tiene a su cargo ocultar y obtener las mediciones del sensor. El módulo *B* oculta e interactúa con la máquina externa por medio de métodos que implementan las reglas de control. Periódicamente, el módulo *B* invoca métodos de *A* para obtener los valores medidos y aplicando las reglas controla la máquina.
 - b) El módulo *A* tiene a su cargo ocultar y obtener las mediciones del sensor. El módulo *B* oculta e interactúa con la máquina externa. El módulo *C* implementa las reglas de control. Periódicamente el módulo *C* invoca métodos de *A* para obtener los valores medidos y luego, según las reglas de control, invoca los métodos de *B* que mueven la máquina
 - c) El módulo *A* tiene a su cargo ocultar y obtener las mediciones del sensor y según esos valores y las reglas de control invoca a métodos de *B* para que este interactúe con la máquina. El módulo *B* oculta e interactúa con la máquina externa.

Soluciones

- a) COMPLETAR.
 - b) COMPLETAR.
 - c) COMPLETAR.
7. ¿Por qué utilizaría DOO y no DTAD?

Solución COMPLETAR.

2. Diseño basado en ocultación de la información (DBOI)

En todos los problemas se pide describir:

- Los ítems con probabilidad de cambio.
- La estructura de módulos (cuando sea interesante).
- La estructura de uso.
- Los módulos 2MIL.
- La guía de módulos.
- La estructura de procesos (cuando sea interesante).

de un DBOI del software que se pide en cada caso.

1. CELDA DE PRODUCCIÓN: La celda consta de un robot y dos cintas transportadoras. El robot cuenta con dos brazos capaces de trabajar cada uno con bulto a la vez. Los bultos viajan sobre las dos cintas transportadoras, cada brazo trabaja sobre una cinta. Cada brazo y cada cinta funcionan independientemente del otro. La tarea del robot es llevar los bultos a una prensa. La prensa sólo puede prensar un bulto a la vez. La prensa tiene un sensor pasivo que indica si está libre o no y uno activo que señala el momento en que se levanta la prensa. El sistema debe indicarle a la prensa cuando prensar y cuando remover el bulto prensado.

Solución

Module	BrazoDercho
exportsproc	recoger() soltar() aPrensa() aCinta()

Module	CintaIzquierda
exportsproc	encender() apagar()

Module	SensorPasivo
exportsproc	libre?():Bool

Module	SensorActivo
exportsproc	listo?(i *f)

Module	Prensa
exportsproc	prensar() remover()

Module	Sistema
imports	BrazoIzquierdo,BrazoDerecho, CintaIzquierda,CintaDerecha, SensorPasivo,SensorActivo, Prensa
exportsproc	iniciar() detener()
private	continuar()

2. MUSEO (Leer con atención al finalizar los requerimientos): Un museo de bellas artes desea instalar un sistema de iluminación automático que mantenga la cantidad de luz dentro de cierto rango que permite a los visitantes apreciar las obras y al mismo tiempo la luz no les produce daños serios.

El museo cuenta con ventanas e iluminación artificial. A las ventanas se les colocan persianas tipo americanas que pueden ser movidas por motores eléctricos de la siguiente manera:

- Las varillas horizontales pueden moverse desde su posición inicial (ventana cerrada) de a un grado hasta que el motor emite una señal que indica que ya no es posible seguir girando, y viceversa.
- La persiana puede elevarse o bajarse de a un centímetro hasta que el motor emite una señal que indica que no es posible seguir bajando o subiendo.

Por otro lado, los artefactos de iluminación artificial son modificados de manera tal que es posible regular su intensidad (aumentándola o disminuyéndola) discretamente.

Al mismo tiempo cerca de cada obra de arte se ha instalado un sensor pasivo que mide la cantidad de luz llega.

En primer lugar, el sistema de software deberá mantener la cantidad de luz sobre cada obra dentro de ciertos límites; y en segundo lugar, el sistema deberá optar primero por utilizar la luz proveniente de las ventanas y si esta no es suficiente deberá hacer uso de iluminación artificial.

- a) Haga un diseño para una obra, una ventana, un sensor de iluminación, y un artefacto de iluminación regulable.
- b) ¿Podría hacer un diseño BOI para el problema más general? Explique su respuesta.

Solución

a)

Module	Control
imports	Varillas, Elevacion, Luz
	SensorVarilla, SensorElevacion
exportsproc	masLuz()
	menosLuz()
private	abierto()
	cerrado()

Module	Elevacion
exportsproc	subirMas()
	bajarMas()

Module	SensorLuz
exportsproc	luminosidad():Int

Module	SensorVarilla
exportsproc	limiteAbierto?(i *f)
	limiteCerrado?(i *f)

Module	Luz
exportsproc	aumentar()
	disminuir()

Module	Sistema
imports	Control, SensorLuz
exportsproc	iniciar()
	detener()
	limiteSuperior(i Int)
	limiteInferior(i Int)

Module	Varillas
exportsproc	abrirMas()
	cerrarMas()

b) COMPLETAR.

3. Una empresa posee una balanza para pesar camiones cargados con materia prima. El camión debe ubicarse más o menos sobre el centro de la balanza para que la pesada sea correcta. Con este fin la empresa instaló cuatro sensores en los vértices de un rectángulo imaginario de forma tal que cuando detectan que el camión está dentro de ese rectángulo, se

debe bajar una barrera detrás del camión. Si el camión rebasa alguno de los laterales del rectángulo se enciende una (de dos) luz ubicada delante del camión que indica qué lado está rebasado.

Cada vez que el camión rebasa alguno de los lados del rectángulo imaginario, el sensor correspondiente emite una señal. Tener en cuenta que puede ocurrir que se rebasa más de un lado al mismo tiempo.

Una vez que el camión está correctamente ubicado y se bajaron las barreras, el chofer debe deslizar una tarjeta magnética que lo identifica. Si la tarjeta es válida, se activa la balanza. Cuando el pesaje finaliza, se debe imprimir un ticket con los datos del conductor y el peso. Luego se levantan las barreras.

Solución

Module	Barrera
exportsproc	subir() bajar()

Module	Balanza
exportsproc	pesar():Int

Module	LuzIzquierda
exportsproc	encender() apagar()

Module	SensorIzquierda
exportsproc	rebasa?(i * f)

Module	Ubicación
imports	SensorIzquierda, SensorDerecha SensorAdelante, SensorAtras
exportsproc	bienUbicado?():Bool iniciar()
private	rebasaIzquierda() rebasaDerecha() rebasaAdelante() rebasaAtras()

Module	Lector
exportsproc	obtenerDatos():Datos

Module	Impresora
exportsproc	imprimir(i Datos)

Module	ControlBarrera	Module	ControlTarjeta
imports	Ubicación, Barrera	imports	Lector, Balanza
exportsproc	controlarBarrera()	exportsproc	controlarTarjeta():Datos

Module	Sistema
imports	ControlBarrera, ControlTarjeta Balanza, Impresora
exportsproc	iniciar() detener()

3. Diseño basado en TADs (DTAD)

En todos los problemas se pide describir lo mismo que en los problemas sobre DBOI , más la estructura de objetos.

1. CELDA DE PRODUCCIÓN: Problema 1 de la sección anterior.

Solución COMPLETAR.

2. MUSEO: Problema 2 de la sección anterior, para todas las obras de arte, ventanas y artefactos de iluminación.

Solución COMPLETAR.

3. PROCESAMIENTO DE IMÁGENES: Un sistema debe procesar un flujo más o menos continuo de imágenes. Las imágenes pueden ser de tres tipos diferentes: color, B/N y 3D. Cada tipo pasa por un proceso diferente aunque puede haber pasos en común.

A las imágenes tipo color se les aplica el siguiente proceso (en el orden indicado): se aumenta el brillo y disminuye el contraste; si el tamaño es menor que cierta cota se la amplía hasta que supere la cota y si es superior a otra cota se la disminuye hasta que esté por debajo; si el rojo o el verde es muy brillante se suavizan.

A las imágenes tipo B/N se les aplica el siguiente proceso (en el orden indicado): si hay zonas muy blancas o muy negras se las suaviza; se hace una copia en negativo de cada imagen. Las originales y sus negativos

siguen los siguientes pasos: se aumenta el brillo y disminuye el contraste; si el tamaño es menor que cierta cota se la amplía hasta que supere la cota.

A las imágenes tipo 3D se les aplica el siguiente proceso: si son en blanco y negro se les aplica el proceso para imágenes B/N; caso contrario el proceso para imágenes tipo color.

Solución COMPLETAR.

- ADMINISTRADOR DE EVENTOS: Diseñe un módulo que que represente un administrador de eventos. Los eventos que administra tienen un nombre, una cantidad de parámetros fija pero (posiblemente) de diferentes tipos. El módulo debe mantener una tabla dinámica que relaciona eventos con procedimientos a invocar.

Solución COMPLETAR.

- MANTENIMIENTO DE CATÁLOGOS: COMPLETAR.

Solución COMPLETAR.

- ESTACIÓN CLIMATOMÉTRICA: Este sistema proporcionará monitorización automática de varias condiciones climatológicas.

Debe medir	También calcula
Velocidad y dirección del viento	Factor de enfriamiento
Temperatura	Temperatura del punto de rocío
Presión atmosférica	Tendencia de la temperatura
Humedad	Tendencia de la presión

El sistema tendrá una forma de determinar la fecha y la hora actuales, de manera que pueda informar de los valores máximos y mínimos de cualquiera de las cuatro medidas principales durante las últimas 24 horas.

El sistema tendrá una pantalla que indicará continuamente todas las medidas así como la fecha y la hora. Mediante el uso de un teclado,

el usuario puede dirigir al sistema para que muestre el máximo o el mínimo en las últimas 24 horas de cualquiera de las medidas principales, junto con la hora en que se dio tal valor. El sistema permitirá al usuario calibrar sus sensores respecto a valores conocidos, y fijar la hora y fecha actuales.

Solución

Module	FechaHora	Module	Teclado
exportsproc	actual():Fecha setear(i Fecha)	exportsproc	tecla(i *f) ultimaTecla():Tecla

Module	Sensor	Module	Sensores
exportsproc	medir():Valor id():Id	exportsproc	sensores():[Sensor] agregar(i Id) quitar(i Id)

Module	Temporizador	Module	Muestrador
exportsproc	setear(i Tiempo, *f)	imports	Sensores
		exportsproc	muestrear():[Valor]

Module	Entrada	Module	Calculador
imports	Teclado, Salida	exportsproc	calcular(i [Valor]):[Valor]
exportsproc	iniciar() procesarTecla()		

Module	Valores	Module	Sistema
imports	Muestrador, Calculador, Temporizador, Salida	imports	Valores, Entrada, Sensores, Salida
exportsproc	obtenerValores() iniciar(i Tiempo)	exportsproc	iniciar(i Tiempo) detener()

Module	Salida	Module	Pantalla
imports	Pantalla	exportsproc	dibujar(i Datos)
exportsproc	redibujar(i [Valor]) vistaPrincipal() vistaIndividual(i Id)		

7. ACCESO A UN CANAL DE NAVEGACIÓN: COMPLETAR.

Solución COMPLETAR.

4. Diseño orientado a objetos (DOO)

En todos los problemas se pide lo mismo que en la sección anterior, más la estructurade herencia (si es interesante).

1. Describa el diseño de un módulo que permita llevar la traza de ciertos objetos de un sistema. Los objetos pueden estar en uno de una cantidad finita de estados, tienen un único estado inicial y los estados están linealmente ordenados. Concretamente se pide que el módulo sea capaz de guardar la información que cada programador desee en cada estado (por ejemplo fecha y usuario) y pueda ser consultado para conocer la traza hasta el momento.

Explique además, con un ejemplo mínimo, cómo usaría ese módulo desde otro cuyas instancias deben ser *tracedas*.

Solución COMPLETAR.

2. PROCESAMIENTO DE IMÁGENES: Problema 3 sección anterior.

Solución COMPLETAR.

3. ADMINISTRADOR DE EVENTOS: Problema 4 sección anterior.

Solución COMPLETAR.

4. MANTENIMIENTO DE CATÁLOGOS: Problema 5 sección anterior.

Solución COMPLETAR.

5. ESTACIÓN CLIMATOMÉTRICA: Problema 6 sección anterior.

Solución COMPLETAR.

6. ACCESO A UN CANAL DE NAVEGACIÓN: Problema 7 sección anterior.

Solución COMPLETAR.

7. COMPLETAR.

Solución COMPLETAR.

8. COMPLETAR.

Solución COMPLETAR.

9. COMPLETAR.

Solución COMPLETAR.