

Índice

1. Estructura de Módulos	1
2. Especificación de Intefaces	2
2.1. Herencia	4
2.2. Módulos genéricos	4
2.3. Patrones de diseño	5
2.3.1. Si es muy largo...	5
2.4. Invocación implícita	6
2.5. Tubos y filtros	7
3. Guía de Módulos: EstacionPeaje	7
3.1. Hardware	7
3.1.1. Barrera	7
3.1.2. BarreraACME	7
3.1.3. BarreraEMCA	8
3.1.4. Impresora	8
3.2. MediosPago	8
3.2.1. HardwareMediosPago	8
3.2.2. SoftwareMediosPago	8
3.3. ControlCarril	9
3.3.1. Control	9
3.3.2. CondicionBajarBarrera	9
4. Estructura de Herencia	9

1. Estructura de Módulos

El criterio aplicado para definir la estructura de módulos es el sugerido por Parnas. Documentamos la Estructura de Módulos usando 2MIL.

Module comprises	EstacionPeaje Hardware MediosPago ControlCarril
-----------------------------------	---

Module comprises	Hardware Barrera BarreraACME BarreraEMCA Impresora
exportsproc	Barrera, Impresora

Hardware exporta **Barrera** para reforzar el hecho de que los clientes deben utilizar únicamente ese módulo y no sus herederos. Notar que esto no significa que no se puedan definir variables de los subtipos de **Barrera** en el código presente fuera del módulo **Hardware**.

De todo lo concerniente a los medios de pago, el resto del sistema sólo precisa acceder a **RecepcionPago** y **Ticket**.

Module	MediosPago
comprises	HardwareMediosPago SoftwareMediosPago TablaPrecios Ticket
exportsproc	RecepcionPago, Ticket

Los dos módulos lógicos que siguen se definen a la espera de otros medios de pago.

Module	HardwareMediosPago
comprises	MaquinaMB

Module	SoftwareMediosPago
comprises	PagoEfectivo RecepcionPago
exportsproc	RecepcionPago

Module	ControlCarril
comprises	Control CondicionBajarBarrera

2. Especificación de Intefaces

Module	BarreraCarrilNorte
exportsproc	subir() bajar() inicializar()
comments	subir() , sube la barrera; bajar() , baja la barrera; inicializar() , pone la barrera en un estado inicial conocido.

Module	ImpresoraCarrilNorte
imports	Ticket
exportsproc	imprimir(i Ticket) inicializar() retiro():Bool
comments	retiro() debe ser llamada para saber si el conductor retiró el ticket; inicialmente retorna false, cuando es invocada y retorna true, a la llamada siguiente retorna false.

Module	MaquinaMBCarrilNorte
imports	Valor
exportsproc	nuevaMB(i *F) denominacion(): Valor cilindroVacio(i *F(i Valor)) capacidadCilindro(i Valor):Int bandejaRetirada(i *F) bandejaInsertada(i *F) entregarMoneda(i Valor) inicializar()

Module	PagoEfectivoCarrilNorte
imports	Valor , MaquinaMBCarrilNorte , Ticket , TablaPreciosCarrilNorte , Monto
exportsproc	hayNuevaMB() noHayCambioDe(i Valor) noHayCambio() hayCambio() pagoEfectivo(): Monto ticket(): Ticket inicializar()
comments	pagoEfectivo() retorna un Monto inválido en tanto no ha finalizado el pago en efectivo y retorna el Monto pagado una vez que se considera que el conductor no entrega más dinero; una vez que retorna un Monto válido a la siguiente invocación retorna uno inválido. Faltan controles para hacer algo útil en caso de que no se pueda dar vuelto y para determinar cuál fue la última moneda o billete insertado.

Module	Ticket
exportsproc	agregarAtributo(i String) agregarValor(i String) primero() siguiente() hayMas():Bool atributo():String valor():String eliminar()
comments	eliminar() borrar el par apuntado en ese momento.

Module	TablaPreciosCarrilNorte
imports	Monto
exportsproc	debePagar(): Monto
comments	El módulo debería tener una interfaz más completa para alterar la tabla de valores.

Module	RecepcionPagoCarrilNorte
imports	Monto , PagoEfectivoCarrilNorte
exportsproc	pago() ticket(): Ticket inicializar()
private	pagoConEfectivo():Bool pagoConTarjetaCredito():Bool pagoConTarjetaDebito():Bool
comments	Para hacer más interesante esta parte del problema suponemos que se han habilitado más de un medio de pago.

Module	ControlCarrilNorte
imports	Ticket , Monto , BarreraCarrilNorte , ImpresoraCarrilNorte , ...
exportsproc	iniciarControl() inicializar()
comments	inicializar() llama a las subrutinas de inicialización de otros módulos.

2.1. Herencia

Module	BarreraACME inherits from Barrera
---------------	--

Module	BarreraEMCA inherits from Barrera
---------------	--

2.2. Módulos genéricos

Generic Module	Lista(X)
imports	X
exportsproc	add(i X) head(): X next(): X more(): Bool ...

A partir de un modulo genérico pueden crearse módulos (no son instancias, son módulos), como se muestra a continuación:

Module	ListaEmp is Lista(Empleado)
---------------	--

Generic Module	Tubo(X)
exportsproc	read():X write(i X)
comments	X puede reemplazarse por array(X) si se quiere una lectura/escritura de mayor longitud.

2.3. Patrones de diseño

Pattern based on because	EstructuraDocumento Composite Cambios previstos: implementación de los distintos elementos que componen un documento, estructura del documento, aparición de nuevos elementos, desaparición de elementos existentes. Funcionalidad: representar la estructura del documento preservando la disposición del texto y los gráficos según los ingresó el usuario, generar y representar visualmente el documento, mantener la relación entre posiciones en la pantalla y elementos de la estructura. Restricciones de diseño: el resto de la aplicación debe acceder a los elementos compuestos (como fila o dibujo) de la misma forma que accede a los elementos simples (como caracter o línea) y lo mismo para los elementos visibles como no visibles.
where	Componente is Glifo Compuesto is Fila Hoja is Caracter Hoja is Rectangulo Hoja is Poligono operacion() is dibujar() operacion() is interseca() operacion() is limites() anadir() is insertar() eliminar() is borrar() obtenerHijo() is hijo() obtenerPadre() is padre() hijos is hijos
comments	<ul style="list-style-type: none">■ obtenerPadre() se menciona implícitamente en la página 154 en el punto 1 de la sección Implementación.

2.3.1. Si es muy largo...

Si la documentación del patrón de diseño sobrepasa el largo de la hoja se puede dividir en secciones como sigue.

Pattern based on because	EstructuraDocumento Composite Cambios previstos: implementación de los distintos elementos que componen un documento, estructura del documento, aparición de nuevos elementos, desaparición de elementos existentes. Funcionalidad: representar la estructura del documento preservando la disposición del texto y los gráficos según los ingresó el usuario, generar y representar visualmente el documento, mantener la relación entre posiciones en la pantalla y elementos de la estructura. Restricciones de diseño: el resto de la aplicación debe acceder a los elementos compuestos (como fila o dibujo) de la misma forma que accede a los elementos simples (como caracter o línea) y lo mismo para los elementos visibles como no visibles.
---------------------------------	--

where	(cont.) Componente is Glifo Compuesto is Fila Hoja is Caracter Hoja is Rectangulo Hoja is Poligono operacion() is dibujar() operacion() is interseca() operacion() is limites() anadir() is insertar() eliminar() is borrar() obtenerHijo() is hijo() obtenerPadre() is padre() hijos is hijos
comments	<ul style="list-style-type: none"> ■ obtenerPadre() se menciona implícitamente en la página 154 en el punto 1 de la sección Implementación.

2.4. Invocación implícita

Module exportsproc announces	Ejemplo
	f(i Int)
	e
	eventWithParameters a b c:Int
callonevent	evento calls sub()
	e a b calls g(b,1)

2.5. Tubos y filtros

Module	Receptor
exportsproc	tiempo(i Int)
inports	datos,paquete:Char
outports	regla,contenido:Char ip:Int

Module	FiltrarRegla
inports	regla:Char
outports	regla:Char

Module	Combinador
exportsproc	establecerCombinacion(i TipoCombinacion)
inports	ip:Int contenido:Char
outports	combinacion:Int

Module	TuboInt is Tubo (Int)
---------------	-------------------------------------

Module	TuboChar is Tubo (Char)
---------------	---------------------------------------

3. Guía de Módulos: EstacionPeaje

El sistema de software para control de la estación de peaje consiste de tres módulos que se describen a continuación.

3.1. Hardware

Este módulo contiene los módulos que deben ser modificados si se reemplaza algún dispositivo de hardware, excepto los relacionados con los medios de pago, por uno similar. Los submódulos de este módulo proveen al resto del sistema un hardware virtual. Los secretos ocultos en estos módulos son las diversas formas en que los dispositivos de hardware deben ser usados.

3.1.1. Barrera

Módulo abstracto que provee únicamente una interfaz para utilizar diferentes marcas o modelos de barreras.

3.1.2. BarreraACME

Oculta las interfaz hardware/software para interactuar con la barrera ACME UpDown 3000.

3.1.3. BarreraEMCA

Ocultar la interfaz hardware/software para interactuar con la barrera EMCA DownUp 0003.

3.1.4. Impresora

Ocultar la interfaz hardware/software para interactuar con la impresora ACME Jetprint 40. Además, ocultar la forma en que se consulta el estado del tique para imprimirlo y el formato en que se lo imprime.

3.2. MediosPago

El módulo lógico **MediosPago** agrupa todos los módulos relacionados con los medios de pago habilitados por la empresa para pagar el peaje. Incluye desde los módulos que ocultan el hardware del sistema hasta los que implementan la política de cobro con uno o más medios de pago. También se incluye la lista de precios y el módulo que implementa el tique que finalmente se le entrega al cliente.

Si la empresa habilita nuevos medios de pago, los módulos que abstraigan esos requerimientos deben agregarse como submódulos de este módulo.

Los secretos que oculta este módulo van desde las interfaces hardware/software con los dispositivos de hardware, hasta la política de cobro pasando por los criterios que se aplican para determinar cuándo ha finalizado el pago por algún medio de pago.

3.2.1. HardwareMediosPago

Aquí se agrupan los módulos que abstraen el hardware que se utiliza para que el conductor pague el peaje, tal como máquinas receptoras de dinero, lectores de tarjetas de crédito/débito, etc. Por el momento, el único módulo físico es el que oculta el hardware de la máquina receptora de dinero.

MaquinaMB Oculta la interfaz hardware/software para interactuar con la máquina receptora de dinero ACME CoinMachine.

3.2.2. SoftwareMediosPago

Este módulo lógico agrupa los módulos que implementan algún medio de pago por sobre los módulos que abstraen el hardware de los mismos. No incluye la lista de precios, ni el tique.

Los secretos que oculta son las condiciones por las cuales se considera que el pago por cada medio de pago ha finalizado, cualquier devolución que deba hacerse al cliente, lo que se incluirá en el tique correspondiente, y la política de cobro (es decir si se permite pagar con varios medios de pago, descuentos, etc.).

PagoEfectivo Este módulo oculta las condiciones que determinan cuándo se considera que el conductor ha finalizado el pago en efectivo insertando monedas o billetes en la máquina correspondiente. También oculta cómo se entrega el vuelto.

RecepcionPago Oculta la política de cobro autorizada por la empresa. Actualmente la política permite pagar con un único medio de pago.

TablaPrecios Oculta las estructuras de datos y algoritmos que implementan la lista de precios así como su ubicación física.

Ticket Oculta las estructuras de datos y algoritmos que implementan el tique que se enviará a la impresora. Se espera que cada medio de pago utilizará este módulo para agregar los datos que sean pertinentes a ese medio de pago.

3.3. ControlCarril

Este módulo incluye los módulos que se encargan de llevar adelante el algoritmo general de procesamiento desde que el conductor se aproxima a la casilla de peaje hasta que se retira. Incluye los módulos que ocultan las distintas condiciones que deben darse para subir y bajar la barrera.

3.3.1. Control

Oculta el algoritmo general de procesamiento.

3.3.2. CondicionBajarBarrera

Este módulo oculta las condiciones (de negocio) para poder bajar la barrera.

4. Estructura de Herencia

Se documenta con 2MIL como se muestra en la Sección [2](#).