Guía de Ejercicios Prácticos

75.07 Algoritmos y Programación III

Practica 1

Torres de Hanoi

a) Implementar el algoritmo de resolución del juego matemático de las torres de Hanoi de n discos de forma iterativa. Se debe utilizar algunas de las colecciones que provee la imagen de Pharo.
b) Refactorizar el código utilizando ahora un algoritmo recursivo.

Reproductor Musical

Implementar la clase ReprounctorMusical que sea capaz de resolver los siguientes mensajes.

- agregarCancion: unaCancion
- agregarCanciones: unaColeccionCanciones
- duracionTotalDeReproduccion
- activarOrdenAleatorio
- activarRepeticion
- reproducir

Este último mensaje debe mostrar en pantalla el nombre de las canciones y la duración de cada una.

Calculadora de Matrices

Programe una clase CalculadoraDeMatrices que permita realizar las siguientes operaciones con matrices: sumar, restar, traza y determinante. Debe permitir operar con matrices de dos dimensiones, de cualquier tamaño. Recuerde que para poder realizar algunas operaciones es necesario que primero verificar que las matrices sobre las que se va a operar sean compatibles.

Practica 2

A partir de esta práctica y por el resto de la guía se deben resolver los ejercicios prácticos de programación adoptando las siguientes reglas que nos plantea TDD:

i. No se puede escribir código de la solución sin una prueba que falle.
ii. No se puede tener más de una prueba en estado fallido a la vez.
iii. No se puede escribir una nueva prueba hasta tanto en código esté debidamente refactorizado.

Dibujando

Dibujar un posible diagrama de clases y uno de secuencia para el siguiente fragmento de código:

unCompas = Compas new.

unCirculo = unCompas dibujarCirculoConRadio:5.

unaCartuchera = Cartuchera new.

unPincel = unaCartuchera getPinceles at:2.

unColor = Rojo new.

unPincel seleccionarColor: unColor.

unPincel pintar: unCirculo.

superficie = unCirculo calcularSuperficie.

Números

Codifique una clase abstracta "Numero" con los siguientes métodos:

- sumar(Numero n): suma a éste número el número "n"
- restar(Numero n): resta a éste número el número "n"
- multiplicarPor(Numero n): multiplica a éste número por el número "n"
- dividirPor(Numero n): divide a éste número por el número "n"
- toString(): (redefinición del método definido en "Object") devuelve un String que representa al número

Más Números

a) Codifique las clases "NumeroEntero", "NumeroReal" y "NumeroComplejo", descendientes de "Numero", que implementen los métodos abstractos de "Numero" mediante aritmética entera, real y compleja respectivamente.
b) Implemente una clase ListaDeNumeros, en la que se encuentre restringida la posibilidad de cargar elementos que no sean Numero (entiéndase que también acepta sus descendientes).
c) Confeccione un diagrama de clases con la lista del punto anterior, y las clases que representan los números.

Lista Circular

Implemente la clase ListaCircular que permite los siguientes mensajes sobre la misma:

- agregarElemento: unElemento
- quitarElemento: elementoAQuitar
- cantidadDeElementos
- estaVacia

Lectura Obligatoria

a) ¿A qué se refiere el artículo “Unit Testing Guidelines” con la frase **Cover boundary cases** ?
b) ¿A qué se refiere el artículo “Unit Testing Guidelines” con la frase **Keep tests independent**?
c) ¿A qué se refiere el artículo “Unit Testing Guidelines” con la frase **Provide negative tests**?

Practica 3

Librería

Una librería vende productos que se dividen en las siguientes categorías: libros, artículos de librería, revistas y periódicos. Las revistas y periódicos tienen una periodicidad que mediremos en cantidad de veces al mes que se publican. Todos los productos tienen un precio asociado. Los artículos de librería tienen además del precio de venta, llevan IVA en la venta. Además el kiosco tiene clientes, de los cuales almacena los datos esenciales incluyendo la dirección, y que productos han comprado en un determinado mes y que suscripciones (a revistas o periódicos) ha adquirido el cliente, los productos que compran los clientes se anotan en su cuenta corriente y se les cobra a fin de mes. El kiosco necesita como funcionalidad saber para un determinado mes cuánto debe cobrarle a cada cliente, y lo mismo para un año entero. Para los clientes registrados en el sistema hay un 5% de descuento sobre todas sus compras y además quienes realizan una suscripción anual existe un 20% de descuento sobre el precio del producto (revista o periódico) al cual se suscriben (este descuento es sobre el precio del producto y no se acumula a otros descuentos).

- Realizar el diagrama de clases correspondiente, detallando los métodos y atributos más importantes.
- Realizar un diagrama de secuencia sobre la obtención de la suma de los importes a cobrarles a los clientes registrados, para un mes determinado.

Juego del Truco

Escriba las clases que considere necesarias para representar un juego de Truco que permita jugar a dos personas. ¿Cómo modificaría su diseño si debiera permitir que una persona juegue contra la máquina?

Puntaje de Tenis

Descripción

Nos centramos sólo en el manejo del marcador de un partido de tenis. La puntuación en el tenis tiene una tendencia del tipo “tira y afloja”.

Conceptos básicos

En un partido de tenis, un jugador empieza con una puntuación de 0. Con cada pelota exitosa, el jugador gana más puntos de la siguiente manera:

$$0 \rightarrow 15 \rightarrow 30 \rightarrow 40$$

Si un jugador llega a los 40 puntos y vuelve a obtener una pelota exitosa, ganará un game. (En la medida que el otro jugador no tenga 40 puntos también).

Si ambos jugadores alcanzan 40 puntos que se conoce como deuce.

Deuce

Una pelota exitosa obtenida en estado de deuce, otorga una ventaja al jugador. Si el jugador contrario marcará nuevamente el marcador vuelve a deuce.

Si un jugador se encuentra en ventaja y marca otra vez, ese jugador gana el game.

Tie Brake

La partida gana cuando un jugador gana 3 sets. Cada set se gana si un jugador llega a 6 games, siempre y cuando tenga diferencia de 2 games con su contrincante. En caso de que ambos lleguen a 6 games ese set se definirá por Tie brake. Aquí la secuencia de puntos es de 1 en 1 y gana el primero que llega a 7 puntos con diferencia de 2. En caso de llegar a 6-6 el ganador deberá estirarse hasta 8-6 y así sucesivamente.

Escriba un programa para manejar cada uno de estos requisitos de puntuación:

- Los jugadores deben ser capaces de sumar puntos.
- El juego debe ser capaz de terminar con un ganador.
- El caso de "deuce" debe ser manejado.
- El caso de “Tie Brake” debe ser manejado también.
- Después de que un juego haya sido ganado, se debe poder determinar al ganador.
- Se debe poder obtener la puntuación actual de cualquier jugador en cualquier momento durante el juego.

Lectura Obligatoria

a) ¿A qué se refiere Ingalls con la frase “Ningún componente en un sistema complejo debería depender de los detalles internos de ningún otro componente” ?
b) ¿Porque Ingalls afirma que “Un programa nunca debería declarar que cierto objeto es ***un*** SmallInteger (entero chico) o un LargeInteger (entero grande)” ?

Practica 4

Batalla de Botes

Para el juego de la Batalla de Botes (similar a Batalla Naval, pero con barcos que ocupan un solo casillero), se requiere una cuadrícula con filas numeradas de 1 a 8 y letras de A hasta la H. Implementar las clases Bote y Tablero con los siguientes métodos, que provea las siguientes funcionalidades:

a) Agregar un bote en un casillero
b) Saber si un casillero está ocupado o no
c) Sacar un bote de un casillero (hundirlo)
d) Reiniciar el tablero con todos los casilleros vacíos.
e) Ubicar 8 botes en lugares aleatorios del tablero

Batalla de Botes II

Extender el juego Batalla de Botes para que soporte tanto los tipos de botes y disparos como las reglas enumeradas a continuación.

Barcos

- Lancha
- Bote a Remo
- Yate

Disparos

- Convencional
- Misil

Reglas

- Los yates solo son afectados solo por los Misiles.
- Se necesitan dos disparos convencionales para hundir una Lancha.

Reversi

Implementar el juego de estrategia Reversi (http://es.wikipedia.org/wiki/Reversi). No se debe implementar la parte gráfica, solo el modelo del mismo. Debe permitir que dos jugadores coloquen sus fichas turno a turno, hasta que el tablero quede lleno o no se puedan colocar más fichas. Al finalizar el juego se debe poder el resultado del juego contando la cantidad de fichas de cada jugador y quien es el ganador.

Go Coreano

Modificar el juego Reversi para adaptarlo a el juego Go Coreano (http://es.wikipedia.org/wiki/Go) Considerar excepciones, en caso de ser necesarias, captura, movimiento, ubicación y suicidio de piezas. ¿Cómo impacto este requerimiento en su código? ¿Debió modificar clases ya existentes o agregar clases nuevas? Si modifico clases, ¿Debió cambiar el comportamiento de los métodos ya existentes o agregar nuevos?

Practica 5

El Ferrocarril

Una empresa de servicios ferroviarios europea define los siguientes conceptos:

Tramo: es un recorrido entre dos ciudades sin cambio de tren, con o sin paradas intermedias. Cada tramo tiene un precio dependiendo del tipo de tren y categoría de pasaje (tipo: tren de alta velocidad, tren nocturno, tren común, etc.; categoría: primera, segunda).

Servicio adicional: es un servicio, no gratuito y optativo, que se puede adquirir junto con un tramo, pero que no se vende separado; por ejemplo: "cucheta coche dormitorio", "cucheta en compartimiento privado", "cena a bordo". No se puede combinar cualquier ítem con cualquier adicional; por ejemplo, sólo se puede contratar cuchetas en trenes nocturnos.

Pasaje: es un derecho de viaje para uno o más pasajeros, para un tramo, o un tramo más uno o más adicionales. Su precio se calcula como suma de los precios de los componentes, por la cantidad de pasajeros. Todos los pasajeros viajan en el mismo tren y en la misma categoría.

Viaje promocional: es un conjunto de pasajes, que pueden incluir varios tramos y que tienen un precio promocional en conjunto; por ejemplo, puede haber un precio especial para la "5 pasajes entre París y Marsella, en TGV, segunda clase, viajando con cambio de tren en Lyon". Los viajes promocionales tienen una vigencia: permanentes, por una estación del año, por día se semana o fin de semana.

Se debe:

- Modelar en UML (diagrama de clases) el problema recién descripto.
- Modelar en UML (2 diagramas de secuencia) el cálculo del precio de un pasaje y el de un viaje promocional.
- Escriba todas las pruebas automatizadas necesarias para probar el comportamiento modelado en el diagrama de secuencia del viaje promocional, usando SUnit.
- Escriba el código Smalltalk que haga funcionar las pruebas del punto anterior, pero sin escribir nada de los métodos de otras clases (dicho de otra manera, los métodos de otras clases debe suponerlos implementados).

Planes de Estudio

Una universidad pública argentina define el siguiente circuito para la aprobación de planes de estudio:

Paso 1: El Consejo Directivo (CD) de la Facultad pide a la Comisión Curricular (CC) de la carrera respectiva (hay una CC por cada carrera de grado, con representantes de docentes, alumnos y graduados) que elabore un plan, dándole un plazo. Dentro de ese plazo, la CC elabora el plan y lo pasa al CD.

Paso 2: Si al cabo del plazo fijado por el CD, la CC no hubiese elaborado un plan, o no hubiese llegado a un acuerdo sobre el mismo, debe pasarle todos los antecedentes al CD. El CD pedirá a la Comisión de Enseñanza (CE) de la Facultad, la adaptación de otro plan preexistente en un plazo menor.

Paso 3: El plan recibido por el CD, proveniente de la CC o de la CE, pasará a consideración del CD, que podrá incorporarle cambios. De haberlos, devolverá el plan a la CC para que opine en un plazo breve.

Paso 4: Una vez que la CC haya opinado, o se le haya vencido el plazo para hacerlo, el CD volverá a analizar el plan, pudiendo aceptar las propuestas de la CC o mantener el plan como estaba.

Paso 5: El plan aprobado por el CD se girará al Consejo Superior (CS) de la Universidad para su aprobación o rechazo. El CS procederá a aprobarlo o devolverlo al CD para que le introduzca cambios. De volver al CD, se repite todo desde el paso 3.

Se pide:

- Haga un diagrama de estados de UML, con todos los eventos que provocan transiciones de estados.
- Modelar en UML (diagrama de secuencia, con objetos y mensajes) la determinación del estado de los planes de estudio de todas las carreras de la universidad.
- Escriba las pruebas automatizadas necesarias para probar el comportamiento modelado en el diagrama de secuencia, usando SUnit, e incluyendo al menos una prueba positiva y una negativa.
- Escriba el código Smalltalk que haga funcionar las pruebas del punto d, pero sin escribir nada de los métodos de otras clases (dicho de otra manera, los métodos de otras clases debe suponerlos implementados).

Transporte Público

Para la implementación de un sistema de control de transporte de pasajeros, se establecen las siguientes definiciones:

Itinerario: un viaje completo a ser realizado por una persona para llegar de un punto a otro del país; un itinerario puede estar compuesto por uno o más tramos.

Tramo: una parte de un viaje entre dos puntos, que se realiza en un solo medio de transporte.

Medios de transporte: pueden ser colectivo, tren urbano, subte, ómnibus interurbano, tren interurbano. Cada medio tiene un sistema tarifario diferente; algunos cobran un precio fijo, otros un valor dependiente de la distancia y otros tienen una tabla de tarifas entre destinos.

Se pide:

- Modelar en UML (diagrama de clases) el problema recién descripto. Tanto en las clases Tramo como Itinerario, deben tener en su comportamiento algún método que permita calcular costo y tiempo de viaje.
- Suponiendo que todas las clases del sistema existen, a excepción de la clase Itinerario, se pide escribir las pruebas automatizadas necesarias para los métodos que calculen el costo y tiempo neto de un itinerario, sabiendo que ambos se obtienen de la simple suma de los costos y tiempos de los tramos.
- Hacer el diagrama de secuencia (UML) del cálculo del costo de un itinerario.
- Escribir, de la clase Itinerario, los atributos y los métodos que permite calcular el costo y el tiempo.

Lectura Obligatoria

- Segun Fowler, ¿Cuáles son los principales beneficios de la integración continua?
- ¿Porque Fowler sostiene que el ámbito donde se realicen las pruebas debe ser una clonación del ambiente real de producción? ¿Es esto siempre posible?