

Primer Parcial

29/04/2017

Tener en cuenta que:

- La duración del parcial es de 3:30 horas.
- Se debe realizar en la computadora, para este fin se deberá realizar un workspace que tendrán que entregar al final el parcial.
- Se entrega un proyecto "Programación 2" que tiene clases de utilidades.
- Culminado las horas del examen se deberá esperar al docente para realizar la copia del trabajo realizado.
- Tenga en cuenta que no se tomaran en cuenta para la corrección clases que no compilen.
- La prueba es SIN MATERIAL.

Preguntas Teóricas :

1. Indique porque un TAD árbol binario de búsqueda, es una estructura con buena eficiencia en el escenario de búsquedas teniendo muchos datos. ¿Existe alguna restricción para que las búsquedas se comporten de forma eficiente?
 2. Describa el TAD Stack y detalle características de su implementación.
 3. Dentro los TAD árboles, se disponen de algoritmos de recorrida que permiten iterar sobre elementos del mismo. Una de estas recorridas es la recorrida por nivel (o por amplitud), la cual se realiza en orden por los distintos niveles del árbol. Así, se comenzaría tratando el nivel 1, que solo contiene el nodo raíz, seguidamente el nivel 2, el 3 y así sucesivamente. Realice un pseudocódigo que permita recorrer por nivel un árbol. Asuma que ya dispone de implementación de los TAD necesarios.
-

Ejercicios Prácticos

Ejercicio 1

Para garantizar la seguridad de los espectáculos deportivos, el estado Uruguayo propone implementar un sistema de control de admisión. Este sistema permitirá usar el derecho de admisión, para negar el acceso al espectáculo a aquellas personas que reiteradas veces han tenido comportamientos fuera de los esperados (por ejemplo comportamientos violentos).

Para ello al ingresar a cada espectáculo se le va a solicitar a la persona su cedula de identidad, un operador verifica en línea si la persona dispone de infracciones (comportamientos no esperados) y si dispone de más de 5 infracciones se le niega la entrada.

Las personas pueden realizar reclamos sobre estas infracciones (si no está de acuerdo con la misma) y un tribunal las atiende y determina si deben anularlas o no. Una misma infracción puede reclamarse múltiples veces, y los tribunales atienden a los mismos en orden de llegada (FIFO).

El sistema se espera que tenga alrededor de 1 millón de personas registradas. Se espera que las personas bloqueadas no superen las 5000 personas, aunque no se puede determinar de antemano un valor exacto.

Con este fin se debe modelar las entidades Persona, Reclamo e Infracción, como también la interfaz SeguridadEstadioMgt y una implementación de la misma SeguridadEstadoMgr que disponga de las siguientes operaciones:

- void **agregarPersona**(String cedula, String nombre, String apellido) throws PersonaYaExiste
 - Agrega una persona identificada por su cédula al sistema.
 - Lanza la excepción PersonaYaExiste en caso de ya haberse registrado la persona.
 - void **agregarInfraccion**(String cedula, long idInfraccion, String descripcion) throws PersonaNoExiste
 - Registra una infracción para la persona con la cédula indicada. El idInfracción identifica la misma y se dispone una descripción que detalla el hecho.
 - boolean **personaEstaHabilitada**(String cedula) throws PersonaNoExiste
 - Consulta si la persona identificada por la cédula está habilitada para ingresar al sistema.
 - void **agregarReclamo**(String cedula, long idInfraccion, String descripcion) throws PersonaNoExiste, InfraccionNoExiste
 - Agrega un reclamo para la infracción y persona indicada. Se debe verificar que tanto la persona como la infracción exista.
 - void **anularInfraccion**(String cedula, long idInfraccion) throws PersonaNoExiste, InfraccionNoExiste
 - Esta operación marca una infracción como anulada. Es importante que se mantenga registro de la infracción (no se borre) pero que la misma no se considere dentro de las 5 permitidas para acceder a los espectáculos.
 - void **imprimirReclamos**()
 - Se debe imprimir en salida estándar (System.out.println) los reclamos recibidos, indicando: cedula, nombre, idEspectaculo, descripción de infracción, descripción de reclamo.
 - Para realizar esta operación, se debe generar una estrategia para iterar sobre la TAD que se seleccione para mantener los reclamos.
-

Consideraciones:

- Se debe seleccionar el TAD más adecuado dado el tipo de problema planteado. Una incorrecta selección no fundamentada, será considerada como incorrecta.
 - Realizar un diagrama de clases en UML en papel de toda la problemática.
 - Realizar test unitarios en JUnit de las operaciones planteadas.
-