

PARCIAL 1 - PARTE 3 – EN EQUIPOS: Ejercicio de programación JAVA

Duración: 60 minutos

Este ejercicio comprende 2 partes:

1. Desarrollo de la funcionalidad especificada más abajo
2. Desarrollo del programa ejecutable con datos de ejemplo provistos y entrega de todo el proyecto Netbeans en la tarea correspondiente de la webasignatura.

ESCENARIO

Una institución de asistencia médica registra a sus cientos de miles de afiliados en una lista, insertándolos en el orden en el que se afilian. El registro y la identificación del afiliado se hace a partir de la cédula de identidad (8 cifras numéricas, sin puntos ni guiones).

Cada afiliado, además de los datos personales básicos (cédula de identidad y nombre y apellido), tiene una lista con el histórico de consultas médicas realizadas y otra lista de consultas médicas a las que está anotado.

Los datos de una consulta son: fecha (de tipo fecha), especialidad (string), médico (número de cédula) y resultado. Resultado es un atributo de tipo entero que puede tener tres valores: 0 si todavía no se ha producido, 1 si el afiliado concurre y 2 si el afiliado no concurre. Al realizarse la anotación, el campo se inicializa en 0, y a la hora de la consulta, el personal de la policlínica registra en 1 si el afiliado concurre, o en 2 si cerrada la hora de la consulta el afiliado no concurre.

Diariamente, a primera hora de la mañana, la contaduría debe recibir una lista de todas las consultas a las que los afiliados no concurren, a efectos de poder realizar el cobro de la misma.

Para ello, el gerente de aplicaciones ha encargado el desarrollo de dos funcionalidades que, cada noche, dada una lista simplemente encadenada de afiliados llamada `padronAfiliados`:

1. para cada afiliado, remueva de la lista las consultas pendientes del día anterior y las inserte en la lista histórica de consultas pasadas, y que además devuelva una lista de los afiliados que, estando anotados, no hayan concurrido a su consulta el día anterior.
2. devuelva una lista de todos los afiliados que estén anotados para consulta de cierta especialidad, ordenada por número de cédula.

PARTE 1: Funcionalidad a desarrollar (vale 50%):

Dada una clase `TPadronAfiliados` que hereda de la clase `TLista`, implementada como una lista simplemente encadenada, desarrollar, y hallar el orden del tiempo de ejecución, dos métodos de `TPadronAfiliados`, de acuerdo a la siguiente firma:

1. `public ILista <IAfiliado> actualizarConsultas ();` // realiza las dos funcionalidades especificadas
2. `public ILista <IAfiliado> consultasEspecialidad (string especialidad);` // debe estar ordenada por cédula

PARTE 2: PROGRAMA (vale 50%).

La clase principal se denomina "Parcial", y tiene su correspondiente método "main". En éste, implementa lo necesario para aplicar los TDA y métodos desarrollados. Se proveen las interfases necesarias para la realización del programa, que no deben ser alteradas. Se debe:

1. Instanciar y cargar los afiliados de la institución en el Padrón, a partir del archivo "padron.txt"
2. Instanciar y cargar las listas de consultas realizadas y de consultas pendientes, a partir de los archivos "históricas.txt" y "agendadas.txt"
3. Aplicar los métodos `actualizarConsultas` (primero) y `consultasEspecialidad` (después) al padrón
4. Emitir por consola el listado a pasar a Contaduría de consultas a las que no concurre, imprimiendo para cada uno: cédula, fecha de consulta.
5. Emitir por consola los afiliados anotados a la especialidad que se escriba en el pizarrón.

ENTREGA: Debes entregar TODO el proyecto Netbeans y los archivos de salida solicitados, en un archivo comprimido "Parcial1.zip" en la tarea "PARCIAL1-PARTE3" publicada en la webasignatura, hasta la hora indicada.