Obligatorio 2

Programación II

Mayo 2009

Fecha de Entrega Preliminar: 2 de Junio del 2009
Fecha de Entrega Final: 22 de Junio del 2009

Planteo del Problema

La empresa **T**ransporte **UR**uguayo **BI**nacional de **NA**ves (por sus siglas T.UR.BI.NA) desea una herramienta capaz de optimizar su flota aérea, capaz de brindarle toda la información posible de sus vuelos entre los diferentes destinos.

Las entidades del sistema que podemos reconocer son las siguientes:

- **Avión:** Del avión se tiene el rendimiento del mismo (km/ltr) de combustible, mas un nombre único provisto por la empresa.
- **Ciudad:** De la ciudad se tienen el nombre y país.
- **Tramo:** Los tramos corresponden a un par ordenado de 2 ciudades (ciudad origen, ciudad destino). Tienen asociada una distancia, y un tiempo estimado de vuelo.

1 Se pide

1.1 Carga/Mantenimiento de información y búsqueda de información

El sistema debe ser capaz de cargar la información anteriormente mencionada a través de un conjunto de métodos especificados en la clase entregada, así como también de realizar búsquedas específicas sobre esos datos cargados. Los métodos principales de carga son:

- Agregar Aviones
- Agregar Ciudades
- Agregar Tramos
- Asignar aviones a tramos (Para un avión, asociar una lista de tramos)
- Quitar un tramo (se quitan sus itinerarios asociados)
- Quitar un avión de circulación (con sus itinerarios asociados)
- Quitar una ciudad (se quitan los tramos que pasan por ella, y los itinerarios asociados a dichos tramos)

1.2 Consultas

A su vez, será necesario realizar un conjunto de consultas sobre todos los datos del sistema:

1.2.1 Listado de los aviones ordenados por rendimiento (de menor a mayor)

1.2.2 Dada una Ciudad Origen y una Destino, obtener el mejor itinerario de vuelo para la compañía (se considera como mejor vuelo para la compañía aquel que implique un menor consumo total)

1.2.3 Dada una Ciudad Origen y una Destino, obtener el mejor itinerario de vuelo para el cliente (se considera como mejor vuelo para el cliente aquel que implique un menor tiempo total)

1.2.4 Encontrar un “Centro de Operaciones”, del cual se minimizan los costos de transporte (se toma como costo de transporte las distancias en kms).

1.2.5 A los efectos de dimensionar las pistas de los aeropuertos, dada una ciudad, indicar los distintos aviones que pueden pasar por ella

2 Controles a realizar y Puntos a Tener en Cuenta

- No se pueden ingresar dos Ciudades con el mismo nombre.
- No se pueden ingresar dos Aviones con el mismo nombre.
- No se pueden ingresar dos Tramos con la misma combinación de CiudadOrigen-CiudadDestino
- Un avión no puede formar parte de un itinerario si no tiene asignado al menos un tramo del mismo
- Pueden haber Aviones sin vuelos asignados.
- Para realizar un vuelo en forma eficiente, el pasajero puede combinar la cantidad de itinerarios/aviones que considere convenientes

3 Clases Entregadas por la Cátedra

3.1 Casos de Testeo

Será entregado un conjunto de casos de testeo mínimos que la aplicación deberá ejecutar en forma satisfactoria. En caso del alumno desear, se les proveerá de un mecanismo de extensión del juego de casos de testeo, para agregar nuevos casos. Junto con los casos de testeo, se entregará un archivo de configuración donde el alumno colocará el nombre de la clase intermedia implementada (ver siguiente sección)

3.2 Clase Intermedia

Será entregada por la cátedra, una clase que contiene todos los métodos que contiene el obligatorio. El estudiante deberá implementar una clase con (al menos) dichos métodos como punto de entrada a su programa. Los casos de testeo invocarán a dichos métodos, sin utilizar los archivos de entrada. Importante: Se calificará positivamente el utilizar dichos métodos también para el procesamiento desde archivos.

4 Entregables

4.1 Carpeta de Análisis

Tanto en la entrega preliminar, como en la final, el alumno deberá entregar una carpeta impresa indicando los lineamientos presentados en el documento "**Plantilla_Informe_v2.0.doc**", ubicado en la Web Asignatura.

Nota: Se descalificará la entrega del código fuente impreso en la carpeta.

4.2 Fuentes del sistema:

1) No se tomaran en cuenta los archivos compilados, así se solicita la no entrega de los mismos, sino de solamente los archivos .java
2) Proyecto conteniendo a las clases del obligatorio
3) Además de los entregables anteriores, se calificará también la **ejecución** del sistema, la **calidad de código** del mismo y **documentación interna** del código fuente, y **testcases agregados**.

4.3 Consideraciones

1) Se tomará como **0 en el Obligatorio** aquel alumno que **no realice la defensa escrita en el día y hora estipulado por los profesores**. En caso de enfermedad, el alumno deberá entregar un certificado justificativo de la misma.
2) Se tomará como **0 en ejecución** un programa que no compile o que dé errores de ejecución o que contenga **virus** o similares.
3) Se recomienda subir una versión (mas allá que no sea la versión final) a la Web de asignatura, para evitar posibles problemas ocasionados por sobrecarga del sistema o similares.
4) **La única identificación dentro del código, o en directorios/archivos será la del ID de grupo** asignado por la cátedra. Se restará calificación a grupos a los que se les encuentre algún otro modo de identificación personal.
5) La documentación del obligatorio debe ser entregada **tanto en formato digital como impresa**.
6) Ante cualquier similitud atípica que se encuentre entre un grupo de trabajos, se podrá proceder a tomar una **defensa oral e independiente** a cada uno de los integrantes

involucrados. En casos en los que se compruebe posibilidad de fraude, se proceder a llevar a cabo las medidas previstas en el reglamento.

7) Los trabajos se realizarán en grupos de a **2 (dos) alumnos**.
8) Es posible que consideraciones hechas sobre la letra del obligatorio sean realizadas a través de la Web de asignatura. En dicho caso, es responsabilidad del alumno mantenerse al tanto de dichos cambios.

5 Aclaraciones adicionales a la especificación del problema:

5.1 Interfaz

No se utilizará interfaz gráfica. El programa se ejecutará desde el editor utilizado en clase.