UNIDAD TEMÁTICA 3: Listas, Pilas, Colas, Orden del Tiempo de ejecución

PRACTICOS DOMICILIARIOS INDIVIDUALES #12

EJERCICIO 1

Una institución educativa ofrece distintos cursos, los que suelen tener diferentes reglas de previaturas. El Coordinador desea contar con funcionalidades que le permitan estimar las cantidades de alumnos que se pueden inscribir a un cierto curso.

Por ejemplo, el curso **"INTEGRADOR 101"** puede nutrirse de alumnos que hayan cursado **"BASICO-ING 1"** o **"BASICO-EMP 1"** indistintamente (o ambos).

Otra situación típica es, por ejemplo, en el curso **"EXIGENTE 102",** que solo puede ser cursado por aquéllos alumnos que hayan aprobado **AMBOS** cursos **"BASICO-ING 1"** o "**BASICO-EMP 1".**

El líder de equipo de desarrollo entiende que estas funcionalidades pueden obtenerse a partir de la abstracción "**CONJUNTO**" y sus correspondientes operaciones típicas: **Unión** e **Intersección**.

Desarrolla los algoritmos para implementar las operaciones de **Union** e **Intersección** sobre el **TDA LISTA**, utilizado para representar un **CONJUNTO**.

- **TConjunto <es una TLista>**
- **TConjunto.Union (TConjunto otroConjunto) : devuelve TConjunto**
- **TConjunto.Intersección (TConjunto otroConjunto) : devuelve TConjunto**

NOTAS:

1. Deben desarrollarse en detalle todos los métodos que se invoquen.
2. Se tendrá en cuenta la eficiencia del algoritmo desarrollado, calificando mejor aquél que tenga mejor tiempo de ejecución o mejor performance general.

EJERCICIO 2

Se requiere implementar la abstracción ***TConjunto*** **-** que deriva de **TLista** – y las operaciones asociadas ***Union e Interseccion*** con la interfaz:

de iConjunto:
public iConjunto Union(iConjunto otroConjunto)

public iConjunto Interseccion(iConjunto otroConjunto)

PROGRAMA

En el método "**main**" de la clase "**Examen**", implementar lo necesario para aplicar los TDA y métodos desarrollados, ejemplificando de la siguiente forma (asumiendo que los cursos pueden ser representados como instancias de **TConjunto**):

1. Instanciar y cargar los datos del curso **"BASICO-ING 1",** (archivo "**basico-ing.txt**") (ver la sección "NOTAS SOBRE ARCHIVOS" al final de este documento por el formato)
2. Instanciar y cargar los datos del curso **"BASICO-EMP 1",** (archivo "**basico-emp.txt**") (ver la sección "NOTAS SOBRE ARCHIVOS" al final de este documento por el formato)
3. Crear un curso "**INTEGRADOR 101**", y asignarle los alumnos que hayan cursado "**BASICO-ING 1**" **y / o** "BASICO-EMP 1", utilizando la operación de iConjunto que corresponda. Guardar el listado de este

curso en el archivo "**Integrador101.txt**". (ver la sección "NOTAS SOBRE ARCHIVOS" al final de este documento por el formato)

4. Crear un curso "**EXIGENTE 102**", y asignarle los alumnos que hayan cursado "**BASICO-ING 1**" **y** "**BASICO-EMP 1**", utilizando la operación de **iConjunto** que corresponda. Guardar el listado de este curso en el archivo **"Exigente102.txt**". (ver la sección "NOTAS SOBRE ARCHIVOS" al final de este documento por el formato)

TEST CASES

Implementa los **Casos de Prueba** necesarios para verificar el correcto funcionamiento de los métodos desarrollados.

NOTAS SOBRE ARCHIVOS:

El departamento de soporte de TI ha provisto las listas de alumnos para "**BASICO-ING1**" y "**BASICO-EMP1**", en las cuales figuran los alumnos que han aprobado estos cursos, y contienen información de la siguiente manera, en cada línea**, SEPARADAS POR COMA**:

COD-ALUMNO // entero

NOMBRE ALUMNO //string

Ejemplo:

1,JOSE

5,JUAN

EL MISMO FORMATO DEBE OBSERVARSE PARA LOS ARCHIVOS DE SALIDA REQUERIDOS