



Universidad Simón Bolívar  
Depto. de Computación y T.I.  
Taller de Algoritmos y Estructuras III (CI-2693)  
Septiembre - Diciembre 2.010

## **Proyecto 0**

### **Programación Orientada a Objetos**

El objetivo de este proyecto es la de una pequeña aplicación utilizando el paradigma Orientado a Objetos.

#### **Enunciado**

La administración de referencias bibliográficas (revistas, libros, etc) suele ser una tarea sumamente tediosa. Para este proyecto Ud debe realizar una aplicación que administre la información de un conjunto de referencias. Su aplicación debe ser capaz de agregar nuevas referencias así como poder consultar y actualizar diferentes aspectos de las existentes.

La aplicación debe leer una secuencia de comandos de un archivo de entrada, realizar la tarea asociada al comando leído e imprimir el mensaje apropiado.

#### **Tipo de Referencias a administrar**

Los tipos de referencias pueden ser muy variados, por simplicidad para este proyecto Ud deberá considerar sólo las siguientes:

- **Libro:** Se refiere a libros de texto
- **Artículo:** Se refiere a documentos publicados en revistas científicas
- **Manual:** Se refiere a manuales de aplicaciones
- **Reporte Técnico:** es un Manual que establece es estado de un proyecto de desarrollo.
- **Tesis:** Libro que contiene la descripción de un trabajo de grado
- **Capítulo:** Se refiere a un capítulo específico dentro de un Libro de texto.
- **Web:** Se refiere a alguna página Web.

#### **Campos comunes**

- autor
- título
- año
- id

#### **Campos específicos por tipo de publicación:**

- **Libro:** editor, Institución
- **Artículo:** Conferencia
- **Tesis:** Institución, Carrera, Tutor
- **Capítulo:** Número de páginas, Título del libro
- **Web:** URL

## Comando permitidos

Su aplicación debe considerar tres comandos:

- **AG:** agregará al catálogo la referencia asociada
- **CO:** Recibe una condición y devuelve toda la información de las referencias que la cumplan. La información debe ser devuelta de forma similar como si se leyera a través del comando AG. Los campos deben ser regresados en orden alfabético.
- **AC:** Recibe una condición y actualiza los valores especificados para las referencias que cumplan la condición. El campo Clave y el Tipo no pueden ser actualizados.

Los campos que deberán ser usados para cada comando serán especificados en la misma línea del comando, cada uno de los campos será especificado con su respectiva palabra clave. Por ejemplo:

1. AG {Tipo = "Libro", Autor = "pepe", Titulo = "El libro escrito por pepe", anio = "2000", clave="pepe2000", editor = "editor de pepe", ISBN = "23-3344"}
  - Debe devolver "OK" en caso de haber podido insertar la referencia en el Catálogo.
2. CO {Autor = "no pepe", anio= "1990"}
  - Devuelve las publicaciones que hayan sido escritas por "no pepe" en el año 1990
3. CO {Tipo = "Libro", Anio="2000"}
  - Devuelve las referencias que sean libros y tengan año de publicación 2000.
4. AC {Clave="pepe2000"}, {Anio="2001"}
  - Las referencias que tengan Clave pepe2000, actualizar al valor del campo Anio por 2001. Sólo debe devolver el número de referencias actualizadas.
5. AC {Tipo="Tesis", Institucion="USB"}, {Institucion="Universidad Simon Bolivar"}

Note que:

- Los nombres de los campos pueden venir en mayúsculas o minúsculas
- Los valores deben ser colocados siempre entre comillas dobles.
- Sólo habrá un comando por línea.
- Cada par (campo, valor) debe estar separado del siguiente por una coma
- El orden de los campos no es importante al momento de leer, sin embargo al devolverlos, éstos deben ser colocados en orden alfabético por el nombre del campo.
- En caso de los dos comandos que reciben una condición, ésta viene especificada justo después del comando.

## Ejecución

Su programa debe poder ser invocado de la siguiente forma:

```
# java Proy0 <archivoOperaciones> <archivosResultados>
```

donde:

- **<archivoOperaciones>** : es el nombre del archivo de donde se leerán todas las operaciones que su aplicación debe procesar (una por línea)
- **<archivosResultados>**: es el nombre del archivo donde su aplicación colocará, uno por línea,

los resultados de cada una de las operaciones. En el mismo orden como se fueron leyendo del archivo de entrada.

Notas:

- Su aplicación debe llamarse `Proy0` para que pueda ser evaluada, de lo contrario su proyecto no será corregido.
- Su aplicación no debe imprimir **ningún** tipo de mensajes por pantalla.
- En caso de ejecución exitosa su aplicación debe retornar el valor 0, y -1 en caso contrario.
- La información a colocar en el archivo debe ser tal cual se haya definida. En caso contrario, el verificador<sup>1</sup> no podrá determinar la correctitud de las operaciones realizadas.
- Se proveerá un pequeño archivo de ejemplo con su respectiva salida. En las corridas, su aplicación será probada en un conjunto más grande de pruebas.
- Se provee un pequeño esqueleto que debe ser usado como base para comenzar el proyecto. Asegúrese que entiende lo que se le entrega y las restricciones, sugerencias, “pistas” ahí especificadas.
- Para este proyecto, se puede asumir que los archivos de entrada son correctos.
- ***Si tiene alguna duda, sugerencia etc, no dude en preguntar con tiempo suficiente.***

**Asegúrese que su aplicación funciona para los casos publicados.**

### Entrega

Para el Martes 5 de Octubre (semana 2) a la 9:30 am usted deberá entregar a su profesor (en el salón de clases), un sobre sellado y debidamente identificado con su nombre, carnet y profesor de laboratorio. Éste debe contener:

- Códigos fuente documentados de todos los tipos implementados. Debe entregarse en un compendio ordenado y engrapado.
- **Un Informe de máximo dos páginas donde muestre mediante un diagrama el diseño de clases de su aplicación.**
- Ud deberá colocar el archivo con su proyecto en Aula Virtual, para lo cual deberá crear el directorio `Proy0` dentro de la carpeta documentos de su grupo. Note que debe estar suscrito a algún grupo en aula Virtual para poder optar a esta opción. En este directorio colocará los archivos con los fuentes de su aplicación. Esto debe hacerlo hasta las 11:59 pm del martes 5 de Octubre.

### Observaciones

En la evaluación del proyecto se tomará en cuenta el diseño de la su solución, el estilo de programación, desempeño, uso de herencia, correcto uso. Proyectos que no compilen serán calificados con cero.

Su código debe respetar la guía de estilo publicada en la sección de documentos de Aula Virtual.

### Referencias

1. [Ortega,Meza] Maruja Ortega y Oscar Meza. Grafos y Algoritmos. Colección Tesis Ciencias Aplicadas. Equinoccio. Caracas, 1993.

---

<sup>1</sup> Un programa que realizará la comparación de las salidas esperadas con las obtenidas por su proyecto.

2. [Ambler] Writing Robust Java Code. Scott W. Ambler Software Process Mentor 2000.
  - Guía de estilo publicada en Aula Virtual [javaCodingStandards.pdf](#)