

*Diseño y Paradigmas de Lenguajes - Año 2014*  
*Práctico Nro. 5*  
*Tipos de Datos Abstractos y Programación Orientada a Objetos*  
*Corresponde a Apunte de TDA y POO*

**Ejercicio 1.**

Considerando que se requiere la implementación del tipo de dato *cola* y que debe ser administrada como tal, es decir, el primer dato en almacenarse debe ser el primero en devolverse, se pide:

- a. Defina el tipo de dato *cola* de valores enteros en lenguaje C, con sus correspondientes operaciones de manipulación:

|                      |   |
|----------------------|---|
| crear cola           | \* crea e inicializa un objeto cola         |
| destruir cola        | \* desasigna el almacenamiento para la cola |
| testeo de cola vacia | \* indica si la cola esta vacia o no        |
| adicionar            | \* adiciona un elemento en la cola          |
| remove               | \* elimina un elemento en la cola           |

- b. Defina nuevamente lo solicitado en el punto anterior, pero utilizando el lenguaje JAVA.
- c. Para cada uno de los puntos anteriores, analice los aspectos de encapsulamiento y ocultamiento de información que provee cada lenguaje.

**Ejercicio 2.**

Escriba un trozo de programa en lenguaje Java en donde se ejemplifique el uso de una *variable de clase* y una *variable de instancia*. Luego, explique cuál es la diferencia entre ambas variables.

**Ejercicio 3.**

Utilizando el lenguaje Java, se pide:

- a. Escribir un programa que defina una clase A, la cual tiene una subclase B, la cual a su vez tiene una subclase C. La clase A debe implementar el método *instancia\_de* el cual será sobrescrito en las clases B y C. Este método muestra el nombre de la clase.
- b. Escribir una clase **Test** en la que se creen objetos de las clases A, B y C. Invocar el método *instancia\_de* de cada clase con su correspondiente instancia.
- c. Suponga ahora que en la clase **Test** se crea un arreglo de 10 posiciones de tipo **Object**, y que dinámicamente se guardará una instancia de A, B o C. Para la asignación dinámica de la instancia a cada posición del arreglo, ud. puede calcular un número aleatorio y dependiendo el rango del mismo seleccionar la instancia de una de las clases, o simplemente solicitar al usuario qué instancia elige guardar en cada posición del arreglo. Luego del llenado del arreglo, invocar al método *instancia\_de*.
- d. Para cada una de las invocaciones de los puntos b. y c. indique el método de qué clase se ejecutará.

**Ejercicio 4.**

Considerando el programa implementado en el ejercicio anterior, ¿qué atributo de método provee Java para forzar la no sobrescritura en las clases B y C del método *instancia\_de* de A?