

## **Tarea Java #7: Arreglos de Caracteres**

\* Nombre de Archivo Java: **Arreglos.java**

\* Ejemplo Nombre Archivo Comprimido:

GutierrezAvilesLuisJ7TSOA0413B.zip

### **Objetivo**

Ejercitar el uso de arreglos

### **Descripción**

Implementar una subrutina que modifique dentro de ella el contenido de un arreglo recibido como parámetro.

### **Requerimientos Generales**

1. Crear un arreglo de caracteres de forma aleatoria con puras letras Mayúsculas.
2. Crear un método que reciba un parámetro arreglo de caracteres.
3. el método copie en dicho arreglo el contenido de una cadena X
4. Esta cadena X contenga puras Mayúsculas y sea menor al arreglo de caracteres.
5. copiar en el arreglo la cadena, tantas veces como quepa completa
6. no usar funciones de copiado del API de Java
7. el método reciba al menos el parámetro arreglo
8. justo después de salir de la subrutina, se imprima el contenido del arreglo de caracteres
9. usar solo variables locales y parámetros
10. imprimir primero el arreglo original, luego el arreglo corregido y por último sólo las celdas modificadas.

### **Requerimientos y Restricciones Complementarios**

- A. Todos los identificadores de variables y constantes tengan nombres claros y significativos de modo que sea fácil saber qué dato almacenarán
- B. Todo identificador debe ser necesario para la operación del programa
- C. Cumplir con las convenciones para nomenclatura de variables y constantes explicados en clase según convenciones Java
- D. Todo valor asignado a un identificador debe ser necesario para la operación de alguna parte del programa
- E. A excepción de las constantes 0 y 1, los cálculos utilicen únicamente variables y constantes simbólicas

## Taller De Sistemas Operativos Avanzados – 2013B – D04

---

- F. A excepción de que un requerimiento lo solicite, no incluir la redacción de los requerimientos en el código fuente
- G. Las líneas de código fuente sean máximo de 100 caracteres
- H. Código fuente indicando como comentario en la parte superior de cada archivo: nombre del(la) alumn@, sección y no. de tarea
- I. Cumplir con las convenciones sobre indentación de código fuente
- J. Código fuente libre de instrucciones anuladas mediante comentarios
- K. El código fuente libre de errores de compilación y advertencias
- L. Código fuente libre de comentarios o impresiones de pantalla que describan el funcionamiento del código
- M. Evitar comparar banderas contra constante numérica o simbólica
- N. Toda instrucción ejecutada sea necesaria según los datos de entrada; por ejemplo, en un programa que validara una fecha en base a día, mes y año, si se trata del mes de Enero, sería inapropiado comprobar si se trata de un año bisiesto, pues esto solo sería necesario para caso del mes de Febrero.
- O. Ejercer la programación modular siempre que sea posible la reutilización de código
- P. Evitar declarar atributos cuyos datos no formen parte de las propiedades que describan la clase en la que se declaren o bien no representen el estado del programa, y que por tanto sea posible una implementación utilizando variables locales y paso de parámetros.
- Q. Todas las clases (y sus atributos) cuenten con un nombre (identificador) claro y significativo a modo que a la lectura sea intuitivo saber para qué se utiliza.
- R. Toda instrucción del programa en ejecución sea ejecutable sin necesidad de recompilar el programa
- S. Cumplir con las convenciones Java (para clases, métodos, etc.).
- T. Todo atributo siempre sea privado, a menos que exista una implementación que por eficiencia justifique un menor ocultamiento de información, pero que igualmente el diseño cumpla con la propiedad de encapsulamiento.
- U. Si incluye comentarios, estos sean solo para Javadoc; en sus métodos, estos sean breves y referentes solo a los parámetros y resultados a obtener, y de las clases solo su propósito; los anteriores en términos de caja negra.
- V. El Código fuente sea libre de instrucciones para suprimir advertencias
- W. Evitar el uso del modificador static, a menos que sea indispensable el uso de atributos de clase y métodos de clase, justificando su uso con comentarios en código

# Taller De Sistemas Operativos Avanzados - 2013B - D04

---

## **Criterios de Evaluación**

- Cumplir con las “Reglas de Operación y Evaluación del Taller de Sistemas Operativos Avanzados”.
- Fecha de asignación: 04 de Septiembre de 2013
- Fecha planeación de entrega: 11 de Septiembre de 2013
- Fecha límite de entrega (Sólo 55% del total): 02 de Octubre de 2013
- Observación: Ninguna
- Calificación en base a cobertura de requerimientos y fecha de entrega en horas clase