

Tarea Java #6: Paso de Parámetros

* Nombre de Archivo Java: **Proceso.java**

* Ejemplo Nombre Archivo Comprimido:

GutierrezAvilesLuisJ6TSOA0413B.zip

Objetivo Comprender las implicaciones del paso de parámetros por referencia

Descripción

Implementar un programa que pruebe la diferencia entre paso por valor y paso por referencia, donde el programa principal se vea como sigue:

```
Public static void main(String[] args){  
  
    Proceso p=new Proceso();  
  
    int a=8;  
    int ar[]=new int[1];  
    ar[0]=a;  
  
    p.valor(a, a);  
    p.referencia(ar,ar);  
  
}
```

Requerimientos Generales

1. Una subrutina que reciba dos parámetros enteros por valor donde:
 - a) modifique el contenido de cada parámetro en su interior
 - b) se modifique un parámetro por línea de instrucción
 - c) mostrar en pantalla el valor en ambos parámetros justo después de que se modifique cada uno de ellos
 - d) llamar a esta subrutina según el código de la descripción previa de main para p.valor
 - e) Mostrar el valor del parámetro una vez que regrese al entorno de main.
2. Una subrutina que reciba dos parámetros enteros por referencia donde:
 - a) modifique el contenido de cada parámetro en su interior
 - b) se modifique el contenido apuntado por cada parámetro, un parámetro por línea de instrucción

Taller De Sistemas Operativos Avanzados – 2013B – D04

- c) mostrar en pantalla el valor en ambas variables apuntadas justo después de modificar cualquiera de los contenidos apuntados
- d) llamar a esta subrutina según el código de la descripción previa de main para p.referencia
- e) Mostrar el valor del parámetro una vez que regrese al entorno de main.

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
- 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) alumno@, 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. Evitar el uso del operador relacional == de una misma variable contra 2 ó más tipos de datos enteros siempre que en su lugar se pueda utilizar la estructura de control “switch”
- P. Programa en ejecución libre de desbordamiento de arreglos
- Q. Todos los métodos tengan nombres claros y significativos de modo que sea fácil saber cuál es su función en el programa.

Taller De Sistemas Operativos Avanzados – 2013B – D04

- R. Ejercer la programación modular siempre que sea posible la reutilización de código
- S. 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.
- T. 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.
- U. Toda instrucción del programa en ejecución sea ejecutable sin necesidad de recompilar el programa
- V. Cumplir con las convenciones Java (para clases, métodos, etc.).
- W. 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.
- X. 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.
- Y. El Código fuente sea libre de instrucciones para suprimir advertencias
- Z. 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

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