

Tarea Java #2: Operadores para manejo de Bits

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

* Ejemplo Nombre Archivo Comprimido:

GutierrezAvilesLuisJ2TSOA0413B.zip

Objetivo

Aprender a utilizar los operadores para manejo de bits y conocer como se ven los datos en la memoria RAM para una variable.

Descripción

Implementar una clase con métodos que impriman la secuencia binaria almacenada en memoria para una variable entera. La tarea trata sobre la impresión en pantalla de los bits que representan un valor entero, mostrándolos tal cual se encuentran en la RAM.

El programa principal contendrá la serie de pruebas para mostrar el resultado según sea el valor y tipo de variable; por ejemplo para variables de tipo int

Bits en la RAM

Pasando el valor int 1:

| 00000000 | 00000000 | 00000000 | 00000001 |

Pasando el valor int 2:

| 00000000 | 00000000 | 00000000 | 00000010 |

Pasando el valor int 128:

| 00000000 | 00000000 | 00000000 | 10000000 |

Pasando el valor int 256:

| 00000000 | 00000000 | 00000001 | 00000000 |

Pasando el valor int 1024:

| 00000000 | 00000000 | 00000100 | 00000000 |

Pasando el valor int 2147483647:

Taller De Sistemas Operativos Avanzados - 2013B - D04

01111111 11111111 11111111 11111111

Nótese que para todos los números positivos, el bit más significativo (el de más a la izquierda) siempre es cero.

Por eficiencia de operaciones aritméticas en la PC, los números negativos se representan mediante "complemento a dos". La clase también deberá soportar números negativos, con lo cual la aplicación debería mostrar en el caso de los enteros:

Pasando el valor int -1:

11111111 11111111 11111111 11111111

Pasando el valor int -2:

11111111 11111111 11111111 11111110

Pasando el valor int -127:

11111111 11111111 11111111 10000001

Pasando el valor int -128:

11111111 11111111 11111111 10000000

Pasando el valor int -2147483648:

10000000 00000000 00000000 00000000

Si a la variable "a" es de tipo byte y le asigna 0x80, el método al que se le pasara la variable imprima:

10000000

...y si "a" fuese de tipo short se imprima:

00000000 10000000

Taller De Sistemas Operativos Avanzados - 2013B - D04

Requerimientos Generales

1. Evitar uso de API de Java
2. Utilizar operadores de corrimiento
3. Declarar en una línea independiente, una variable de un byte de capacidad
4. Asignarle a la variable del punto anterior una combinación de bits, usando un valor en notación hexadecimal vía código fuente
5. Implementar método para imprimir en pantalla la serie de bits (0s y 1s) para el tipo de dato de 1 byte
6. Implementar método para imprimir en pantalla la serie de bits (0s y 1s) para el tipo de dato de 2 bytes
7. Implementar método para imprimir en pantalla la serie de bits (0s y 1s) para el tipo de dato de 4 bytes
8. Ejercer sobrecarga de métodos respecto a los 3 anteriores
9. Antes de imprimir la secuencia de bits imprimir el caracter pipe y un espacio.
10. Después de cada ocho bits impresos imprimir un espacio, el caracter pipe y un espacio.
11. El programa imprima correctamente los bits para números positivos.
12. El programa imprima correctamente los bits para números negativos.
13. El método main contenga en hardcode las pruebas citadas en la descripción
14. Es optativo un método para tipo de 8 bytes

PUNTOS EXTRAS (50%)

- Implementar conversiones en tipo de dato Long (8 bytes)

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. Todo valor asignado a un identificador debe ser necesario para la operación de alguna parte del programa
- D. A excepción de las constantes 0 y 1, los cálculos utilicen únicamente variables y constantes simbólicas

Taller De Sistemas Operativos Avanzados – 2013B – D04

- E. Código fuente indicando como comentario en la parte superior de cada archivo: nombre del(la) alumno@, sección y no. de tarea
- F. Código fuente libre de instrucciones anuladas mediante comentarios
- G. El código fuente libre de errores de compilación y advertencias
- H. Código fuente libre de comentarios o impresiones de pantalla que describan el funcionamiento del código
- I. Programa en ejecución libre de desbordamiento de arreglos
- J. Todos los métodos tengan nombres claros y significativos de modo que sea fácil saber cuál es su función en el programa.
- K. Ejercer la programación modular siempre que sea posible la reutilización de código
- L. 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.
- M. 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.
- N. El Código fuente sea libre de instrucciones para suprimir advertencias
- O. 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

- Los establecidos en las “Reglas de Operación y Evaluación” del Taller de Sistemas Operativos Avanzados.
- Fecha de asignación: 28 de Agosto del 2013
- Fecha planeación de entrega: 04 de Septiembre del 2013
- Fecha límite de entrega (Sólo 55% del total): 25 de Septiembre del 2013
- Observación: Ninguna
- Calificación en base a cobertura de requerimientos y fecha de entrega en horas clase