

Tarea Java #1: Ocultamiento de Información y Encapsulamiento

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

* Ejemplo Nombre Archivo Comprimido:

GutierrezAvilesLuisJ1TSOA0413B.zip

Objetivo

Aplicar el concepto de encapsulamiento para ocultar los detalles de implementación de clases.

Descripción

La tarea trata acerca de la implementación de una clase MiFecha, la cual siempre mantenga un estado consistente en la representación de cualquier fecha posible de la realidad.

Años

Cualquier año sea válido, esto es, en el rango de los enteros con ó sin signo.

Meses

El número correspondiente a cada mes a validar es como sigue:

Mes	Número de mes
Enero	1
Febrero	2
Marzo	3
Abril	4
Mayo	5
Junio	6
Julio	7
Agosto	8

Taller De Sistemas Operativos Avanzados - 2013B - D04

Septiembre	9
Octubre	10
Noviembre	11
Diciembre	12

Días

Los días válidos para cada mes son como sigue:

Mes	Días
Enero	Del 1 al 31
Febrero	Del 1 al 28 para todos los años 29 para los años bisiestos
Marzo	Del 1 al 31
Abril	Del 1 al 30
Mayo	Del 1 al 31
Junio	Del 1 al 30
Julio	Del 1 al 31
Agosto	Del 1 al 31
Septiembre	Del 1 al 30
Octubre	Del 1 al 31
Noviembre	Del 1 al 30
Diciembre	Del 1 al 31

Taller De Sistemas Operativos Avanzados – 2013B – D04

Años bisiestos

Un año bisiesto se presenta cada 4 años contados a partir del año cero, esto es:

0, 4, 8, 12...etc. ...son años bisiestos.

Hay una excepción para los años bisiestos, esto es cada 100 años:

100, 200, 300, 500, 600, 700...etc ...no son años bisiestos.

Cómo se habrá notado en la serie anterior, hay una excepción a la excepción, esto es que cada 400 años:

400, 800, 1200...etc. ...sí son años bisiestos.

Requerimientos Generales

MiFecha
<pre>-dia: int -mes: int -año: int -verificaDia(in d:int): bool +dameDia(): int +dameMes(): int +dameAño(): int +fijaDia(in d:int): bool +fijaMes(in m:int) +fijaAño(in a:int)</pre>

1. Crear el archivo *fuentes* Encapsulamiento.java para aplicación de consola conteniendo al método *main*
2. Codificación de la clase MiFecha en el mismo *archivo* y según se ilustra en el diagrama de clase UML
3. Inicializar todos los atributos con datos válidos por medio de un constructor sin parámetros
4. Declarar y definir todos los métodos *dame()*
5. Declarar y definir todos los métodos *fija()*
6. Incluir al menos el método *verifica()* mostrado (se puede optar por otra forma en nombre y parámetros, pero igualmente privada)
7. Es optativo y deseable que los otros métodos *fija* también sean booleanos.
8. Contemple que un año bisiesto es múltiplo de 4, no múltiplo de 100 pero sí múltiplo de 400
9. Programarla lo más completa y sintética posible de forma que siempre represente en sus atributos una fecha válida

Taller De Sistemas Operativos Avanzados – 2013B – D04

10. Validación en cada uno de los métodos fija para que los atributos en conjunto representen siempre una fecha correcta sea cual fuere el orden posible de invocación a los métodos *fija()* desde *main*
11. Si al tratar de cambiar algún atributo mediante un método *fija()*, la nueva fecha no sería válida, los atributos permanezcan sin modificación.
12. Evitar imprimir mensajes a consola desde métodos al interior de la clase (el método *main* no forma parte de la clase)
13. Codificar en el ámbito del método *main* lo siguiente:
 - a) Declaración de *variable* de instancia del tipo *MiFecha* en una línea independiente
 - b) Instanciación de un *objeto* del tipo *MiFecha* sobre la *variable* del punto anterior
 - c) Impresión en consola del contenido de cada uno de los atributos, uno por línea
 - d) Crear menú para poder hacer cambio de día, mes o año.

PUNTOS EXTRAS

Se calificará con un 50% más a aquellos que en vez de utilizar impresión a consola realicen esta tarea 1 utilizando ventanas.

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. A excepción de que un requerimiento lo solicite, no incluir la redacción de los requerimientos en el código fuente
- D. Código fuente indicando como comentario en la parte superior de cada archivo: nombre del(la) *alumn@*, sección y no. de tarea
- E. Cumplir con las convenciones sobre indentación de código fuente
- 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. 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

Taller De Sistemas Operativos Avanzados – 2013B – D04

una implementación utilizando variables locales y paso de parámetros.

- 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: 21 de Agosto del 2013
- Fecha planeación de entrega: 28 de Agosto del 2013
- Fecha límite de entrega (Sólo 55% del total): 18 de Septiembre del 2013
- Observación: Ninguna
- Calificación en base a cobertura de requerimientos y fecha de entrega