FACULTAD DE INGENIERÍA

FORMATO
PRÁCTICAS DE LABORATORIO

UNIVERSIDAD AUTÓNOMA DE CAMPECHE

NOMBRE DEL PROFESOR: Ing. Juan Antonio Chuc Méndez
NOMBRE DE LA PRÁCTICA: Acceso a miembros
PRÁCTICA NÚM. [3]

LABORATORIO:	CENTRO DE CÓMPUTO
MATERIA:	LENGUAJE DE PROGRAMACIÓN II
UNIDAD:	DOS
TIEMPO:	2HRS

1	**Introducción**
Los objetos proporcionan encapsulamiento. En general, sucede cuando algo es envuelto en una capa protectora. Cuando el encapsulamiento se aplica a los objetos, significa que los datos están protegidos, "ocultos" dentro del objeto. A cambio, el resto del programa no puede acceder de manera directa a los datos de un objeto; lo tiene que hacer con ayuda de los métodos de un objeto.	

2	**Objetivo**
Comprender el principio del encapsulamiento, aprender la manera en que se implementa, y el mecanismo para proveer acceso a miembros que han sido encapsulados.	

3	**Fundamentos**
El encapsulamiento es uno de los pilares sobre el cual se basa el paradigma de la Programación Orientada a Objetos. Permite hacer que una clase se preocupe únicamente de su propio funcionamiento, haciendo que una tarea compleja pueda dividirse fácilmente en tareas más sencillas. Dado que una clase es responsable únicamente de la forma en que ella misma funciona, otras clases pueden utilizarla sin preocuparse por cómo la clase realiza sus operaciones, sino que simplemente reciben el resultado y ya. En la práctica, se procede a ejemplificar el uso y funcionamiento de los diferentes tipos de modificadores de acceso, así como la manera de proveer acceso a miembros que han sido encapsulados.	

4	Procedimiento

A) **Herramientas y materiales del alumno**

1. Computadora Personal
2. Plataforma de Software para el lenguaje Java.

B) **Desarrollo de la práctica**

1. **Abre** el archivo con código fuente **Motor.java.**
2. **Examina** la clase. La clase cumple con algunos de los principios del encapsulamiento, pero está incompleta.
3. **Compila** la clase **Motor.java.**
4. **Abre** el archivo con código fuente **AutomovilFord.java.**

AutomovilFord es una modificación de nuestra clásica clase Automovil, pero que ha sufrido algunas modificaciones: por ejemplo, la marca ahora solamente puede ser "Ford" (de ahí el nombre de la clase) y se sustituyó la variable booleana motor por una referencia a la clase 'Motor'.

5. **Observa** que a diferencia de la clase 'Motor', la clase 'AutomovilFord' no cumple una sola de las reglas del encapsulamiento.
6. **Compila** la clase 'Automovil'. Un montón de errores surgieron. No temas, ¡pronto los arreglaremos!
7. Dentro de la clase 'Motor' **añade** métodos de acceso (setters o getters, o ambos) para las variables que se necesitan utilizar en la clase AutomovilFord (encendido y revoluciones).
8. De vuelta a la clase 'AutomovilFord', **sustituye** el intento de acceso a las variables privadas de 'Motor' por los métodos de acceso correspondientes. 4 de los 7 errores deben haber desaparecido.

'AutomovilFord' intenta utilizar métodos que 'Motor' declaró como privados, pues comprometen su funcionamiento. 'Motor' debe ser el único responsable de su funcionamiento, y sin embargo, 'AutomovilFord' intenta hacer funcionar el motor… ¡en el orden incorrecto! Hagamos pues, que el único responsable del funcionamiento del motor sea la propia clase 'Motor'.

9. Crea el siguiente método en la clase 'Motor'

```
public void revolucionar() {
  if(encendido) {
    prepararMezcla();
    comprimirMezcla();
    encenderMezcla();
  }
}
```

10. **Modifica** el método acelerar() de la clase 'AutomovilFord'. Borra los tres métodos que intentaban hacer funcionar al motor y sustitúyelas por una llamada al método revolucionar() que creaste en el paso anterior.
11. Modifica el método frenar(int) de la clase 'AutomovilFord' para que incluya una llamada al método de acceso para el atributo 'revoluciones' que creaste en el paso 7, y que establezca su valor en la cantidad proporcionada como argumento.
12. Compila ambas clases: 'AutomovilFord' y 'Motor'.
13. Abre el archivo con código fuente 'TesterFord.java'.
14. **Compila** y **ejecuta** la clase TesterFord. El AutomovilFord funciona correctamente, ya que el Motor se hizo responsable de su funcionamiento.

Fin de la práctica.

5 Resumen

1. A las variables y métodos de una clase se les conoce como miembros de clase.
2. Si se declara un miembro como público, todas las demás clases podrán acceder a él.
3. Si se declara un miembro como privado, solamente puede ser accedido a este dentro de la misma clase.
4. El operador **this** es la manera en la que se le dice al compilador de Java que nos estamos refiriendo a un miembro dentro de nuestra misma clase.

6 Ejercicios

- En tu libreta, escribe responde a las siguientes preguntas:
  - En Programación Orientada a Objetos ¿Qué es el encadenamiento?
  - Escribe 3 clases donde puedas demostrar cómo realizar el encadenamiento. Puede ser del tema que tu prefieras. Una de esas clases debe llamarse TesterEncadenamiento, con un método main que ponga a prueba este concepto.

7 Referencias

- Dean, J. S., & Dean, R. H. (2009). Introducción a la programación con Java. México: Mc Graw Hill.
- Apuntes del profesor.

8	Firma	Firma y Fecha de recibido
	_____ Nombre y firma del docente.	