



UNIVERSIDAD AUTÓNOMA DE CAMPECHE

NOMBRE DEL PROFESOR: Ing. Héctor Manuel Quej Cosgaya**NOMBRE DE LA PRÁCTICA:** Fundamentos de Programación Orientada a Objetos**PRÁCTICA NÚM.** [7]

LABORATORIO:	Centro de Ingeniería Computacional
MATERIA:	Lenguaje de Programación I
UNIDAD:	Subcompetencia IV
TIEMPO:	2 horas

OBJETIVO:

Comprender los conceptos fundamentales de la Programación Orientada a Objetos: objeto, clase, atributo y método; aprender la estructura básica de una clase, diferentes tipos de atributos y métodos

MARCO TEÓRICO:

El paradigma de la Programación Orientada a Objetos ha venido ganando fuerza en los últimos tiempos, dado su naturalidad para modelar sistemas, entidades y situaciones del mundo real. Un programa Orientado a Objetos permite comprender más fácilmente su funcionamiento, pues modela una entidad del mundo real y se comporta exactamente de la misma manera. En la práctica, se procede primero a crear una clase Automovil utilizando el entorno de desarrollo interactivo BlueJ y se expone el funcionamiento básico de los objetos y de un programa orientado a objetos.

LISTA DE MATERIALES:

- Java SDK
- Entorno de Desarrollo Interactivo BlueJ

EQUIPO DE LABORATORIO:

- Computadora Personal

DESARROLLO DE LA PRÁCTICA:

1. Abre el **Entorno de Desarrollo Interactivo BlueJ**. Normalmente hay un acceso directo en el escritorio para él. De lo contrario, lo encontrarás en **Inicio → Todos los programas → BlueJ → BlueJ**.
2. Haz clic en el menú **Project → New Project** para crear un nuevo proyecto de **BlueJ**.
3. Selecciona una ubicación para tu proyecto, y asígnale un nombre. Haz clic en el botón



UNIVERSIDAD AUTÓNOMA DE CAMPECHE

'Create'. En la ubicación que seleccionaste, se creará una nueva carpeta con el nombre de tu proyecto, que contiene dos archivos: un archivo de texto README.TXT y un archivo de proyecto de **BlueJ**.

4. Del lado izquierdo de la aplicación, haz clic en el botón 'New class...'. Aparecerá una pequeña ventana de configuración. Establece el nombre de la clase en 'Automovil', y deja el tipo de clase en 'Class'. Presiona el botón 'Ok' para crear tu nueva clase.

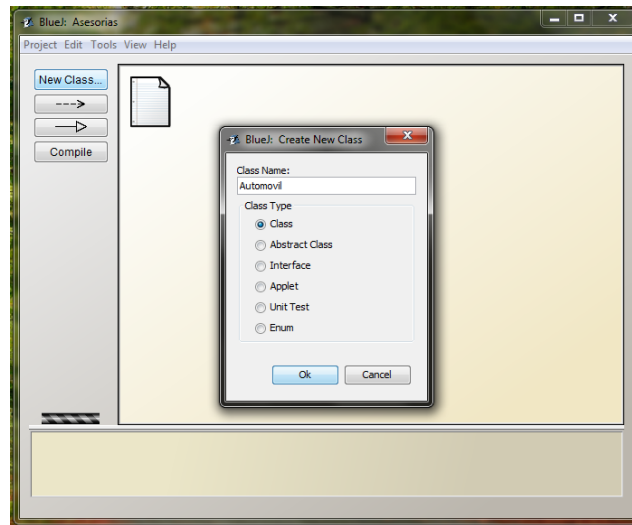
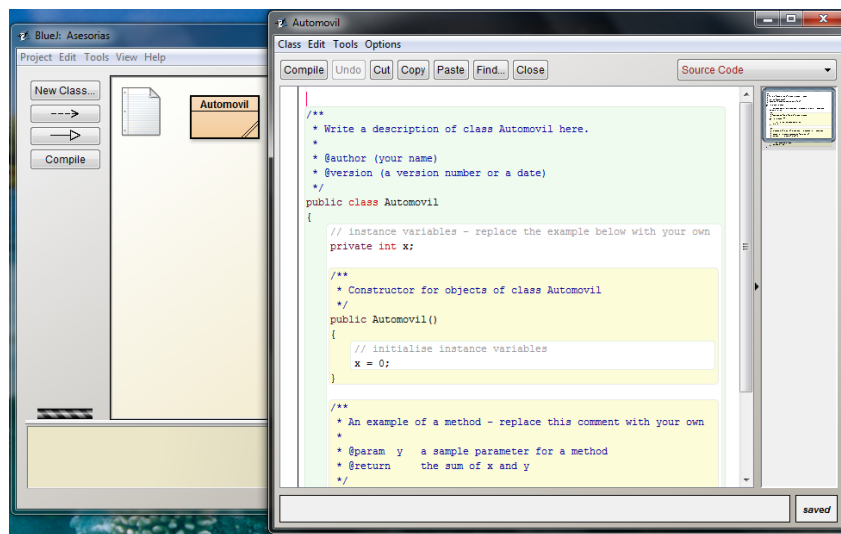


Fig. 1 – Creación de una clase en BlueJ

5. En el panel central de **BlueJ**, ahora aparece un pequeño rectángulo naranja con el título 'Automovil': es la representación de la clase que acabamos de crear. Haz doble clic sobre ella para abrir el editor de texto de **BlueJ**.





UNIVERSIDAD AUTÓNOMA DE CAMPECHE

6. En el editor de texto de **BlueJ**, borra la declaración de la variable entera 'x'. Luego añade las siguientes **variables de instancia**:
 - a. int velocidad = 0;
 - b. boolean motor = false;
 - c. String color;
 - d. String modelo;
7. Inmediatamente después de las **variables de instancia** que declaraste en el paso anterior, añade las siguientes **variables de clase**:
 - a. static String marca;
 - b. static int numeroDeAutos = 0;
8. Modifica el **constructor** de la clase Automovil para que reciba los siguientes **parámetros**:
 - a. String fabricante
 - b. String tipo
9. Borra la instrucción del **constructor** de la clase Automovil que inicializaba a la variable x. En su lugar, escribe las siguientes instrucciones:

```
modelo = tipo;
marca = fabricante;
numeroDeAutos++;
```

10. Borra el **método** de ejemplo que crea **BlueJ** para nuestra clase. En su lugar, escribe los siguientes **métodos de instancia**:

```
public void encenderMotor() {
    motor = true;
    System.out.println("Motor encendido");
}

public void apagarMotor() {
    motor = false;
    System.out.println("Motor apagado");
}

public void acelerar(int cantidad) {
    if(motor){
        velocidad += cantidad;
        System.out.println("Vamos a " + velocidad + " kph");
    } else {
        System.out.println("El motor está apagado!");
    }
}
```



UNIVERSIDAD AUTÓNOMA DE CAMPECHE

```
public void frenar() {  
    velocidad = 0;  
    System.out.println("Nos hemos detenido!");  
}  
  
public int verVelocidad() {  
    return velocidad;  
}
```

11. Después de los **métodos de instancia** que acabas de agregar a nuestra clase, añade los siguientes **métodos de clase**:

```
public static int verNumeroDeAutos() {  
    return numeroDeAutos;  
}  
  
public static void pintar(Automovil auto) {  
    auto.color = "Negro";  
}
```

12. En la parte superior del editor de texto de **BlueJ**, haz clic en el botón **'Compile'**. Si no hay errores en tu código, el editor mostrará un pequeño mensaje de confirmación en la barra inferior.

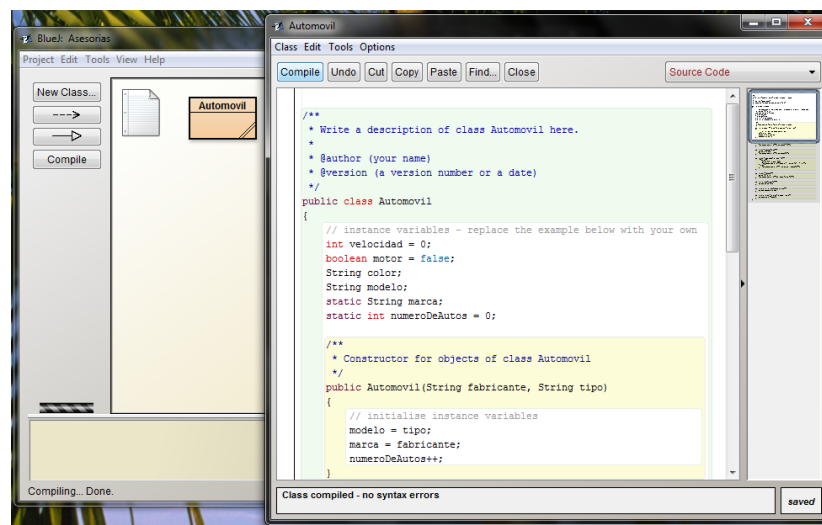


Fig. 3 – La clase Automovil completa y compilada.

13. Cierra el editor de texto de **BlueJ**.
14. En la pantalla principal de **BlueJ**, haz clic derecho sobre nuestra clase 'Automovil'. Aparecerá un pequeño menú contextual con las opciones que podemos realizar. Haz clic en la opción 'new Automovil (String fabricante, String tipo)'.
15. Tras hacer clic en la opción anterior, aparecerá un pequeño cuadro de diálogo. En él



UNIVERSIDAD AUTÓNOMA DE CAMPECHE

tenemos que establecer los **argumentos** que le proporcionaremos al **constructor** de nuestra clase `Automovil` para que cree un nuevo **objeto**. Proporciona los siguientes **argumentos** para crear tu objeto `Automovil`:

- a. Nombre de instancia: `coche1`
 - b. Fabricante: "Ford"
 - c. Tipo: "Fusion"
16. Haz clic en 'Ok' para cerrar el cuadro de diálogo de creación de objetos. Observa que en el panel inferior de **BlueJ** ahora aparece un rectángulo rojo que contiene las palabras "coche1: Automovil". Este es la *representación* del **objeto** que acabamos de crear.

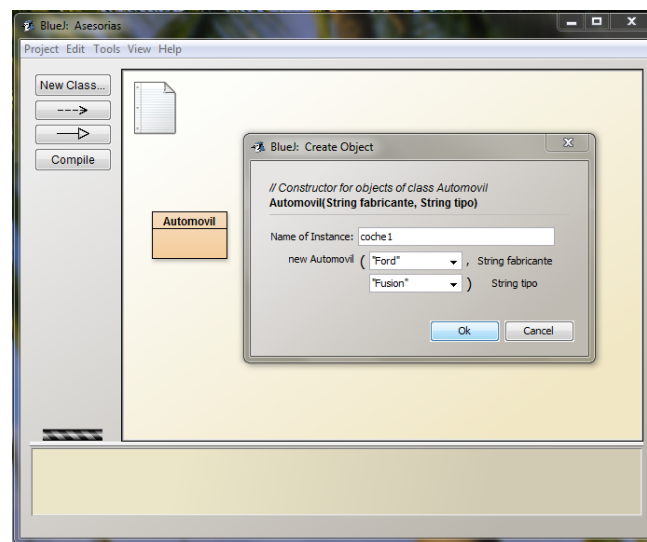


Fig. 4 – El cuadro de diálogo de creación de objetos en BlueJ.

17. Haz doble clic en el rectángulo rojo que representa al **objeto** `coche1`, se abrirá una pequeña ventana con los **atributos de instancia** del objeto `coche1`. Verifica que su **atributo** `modelo` recibió el valor "Fusion" que nosotros indicamos como **parámetro** a la hora de su creación. Todos los demás **atributos** de instancia de `coche1`, salvo `color`, tienen un valor asignado.
18. Dentro de la ventana que muestra los atributos del objeto `coche1`, haz clic en el botón "Show static fields". Aparecerá otra ventana, esta vez, con las **variables de clase** de `Automovil`. Observa de nuevo que la **variable de clase** `marca` tiene asignado el valor "Ford" que nosotros especificamos durante la creación del **objeto** `coche1`.
19. Cierra ambas ventanas inspectoras.
20. Haz clic derecho sobre el rectángulo rojo que representa al objeto `coche1`, y selecciona la opción "void encenderMotor()". Acabas de **llamar al método** del mismo nombre del objeto `coche1`. Aparecerá una pequeña ventana terminal con el mensaje impreso por el **método** `encenderMotor()`.

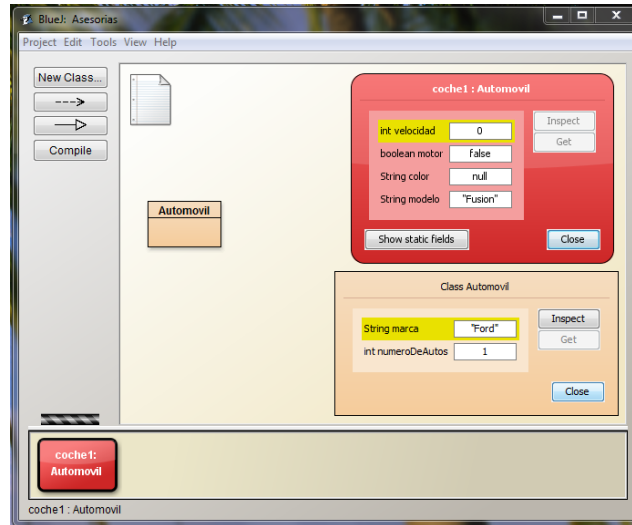


Fig.5 – El Inspector de atributos de BlueJ mostrando los atributos del objeto coche1 y de la clase Automovil

21. Haz clic derecho sobre el rectángulo rojo que representa al objeto coche1 de nuevo, y selecciona la opción “void acelerar(int cantidad)”. Aparecerá un cuadro de diálogo donde se te solicitará que introduzcas el **argumento ‘cantidad’** que el **método acelerar(int cantidad)** necesita. Introduce cualquier número en el cuadro y haz clic en el botón **‘Ok’**. La pantalla con la terminal ahora muestra el mensaje de confirmación enviado por el método **acelerar(int cantidad)**.

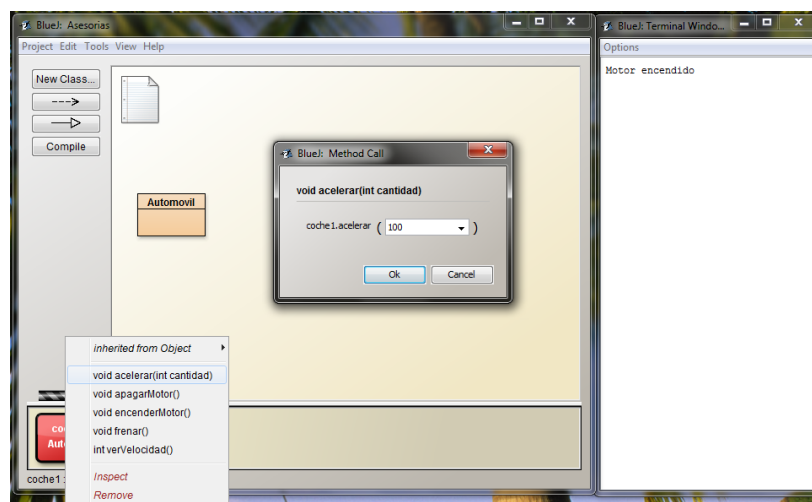


Fig. 6 – Terminal de salida, invocación y paso de argumentos a un método en BlueJ

22. Haz clic derecho en el rectángulo rojo que representa a nuestro objeto coche1, y selecciona la opción “int verVelocidad();” Aparecerá una pequeña ventana con el **valor de retorno** que regresa el **método verVelocidad()**. Inspecciona la ventana y



UNIVERSIDAD AUTÓNOMA DE CAMPECHE

ciérrala cuando termines.

23. Haz clic derecho en el rectángulo naranja que representa a nuestra clase Automovil, y selecciona la opción “void pintar(Automovil auto)”. Aparecerá la ventana para solicitarte el **argumento** que el **método** pintar(Automovil auto) necesita. La razón por la que este **método** no aparece si haces clic derecho en el rectángulo rojo que representa a coche1 es por que pintar(Automovil auto) es un **método de clase**.
24. Proporciona como **argumento** al método pintar(Automovil auto) la **instancia** de la clase Automovil que ya tenemos, es decir, coche1. Haz clic en el botón ‘Ok’.
25. Nuevamente haz doble clic en el rectángulo rojo que representa a nuestro objeto coche1 para abrir el inspector de **atributos**. Observa que el **método** pintar(Automovil auto) ha establecido el **atributo** color de nuestro coche1 en “Negro”.

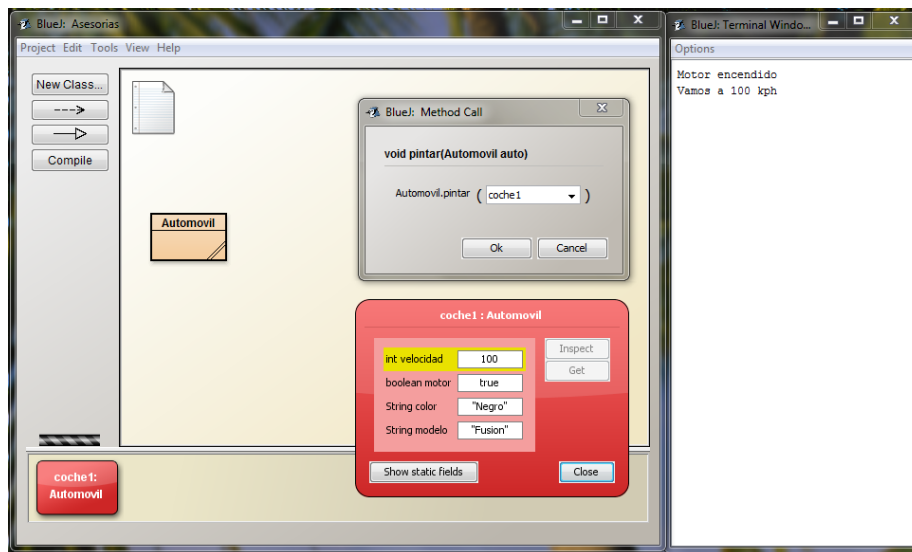


Fig. 7 – Invocación de un método de clase en BlueJ y su resultado

26. Haz clic en el menú **Project** y selecciona la opción “**Save**” para guardar tu proyecto.
27. Cierra **BlueJ**.

Fin de la práctica

RETROALIMENTACIÓN:

- Elabora un proyecto de BlueJ que modele a una máquina dispensadora de refrescos. Debes plasmar todas las operaciones de la máquina, como recibir dinero, especificar el producto que desea el usuario, dar cambio, entregar el producto, etc.



UNIVERSIDAD AUTÓNOMA DE CAMPECHE

RECOMENDACIONES ADICIONALES:

- Lee el capítulo 6 del Dean (Programación Orientada a Objetos)
- Investiga más acerca de las funcionalidades de BlueJ.

BIBLIOGRAFÍA:

- Dean, J. S., & Dean, R. H. (2009). Introducción a la programación con Java. México: Mc Graw Hill.
- Roberts, Simon; Heller Philip y Ernest, Michael (1999). The Complete Java 2 Certification Study Guide. Alameda, California: SYBEX.
- Apuntes del profesor.