



Carátula para entrega de prácticas

Laboratorio de computación salas A y B

Profesor: Ing. Ricardo Martínez Moreno

Asignatura: Programación Orientada a Objetos

Grupo: 7

No de Práctica(s): 4

Integrante(s): Ahuactzin Lopez Jorge Daniel, Landázuri Brambila

Álvaro Ulises, Marcial Hernández Claudia Sarahi,

Medina Molina Fernando

Semestre: 2018-1

Fecha de entrega: Domingo 10/Septiembre/2017

Observaciones:

CALIFICACIÓN: _____

Ejercicio 1

Código

```
Waffles@Ulises-MacBook-Air: [~/Documents/UNAM/3° Semestre/P00/P4_P00/code/P4 (master)]$ cat Fabrica.java
public class Fabrica{

    private String nombre;
    private String direccion;

    public String getNombre(){
        return this.nombre;
    }

    public void setNombre(String nombre){
        this.nombre = nombre;
    }

    public String getDireccion(){
        return this.direccion;
    }

    public void setDireccion(String direccion){
        this.direccion = direccion;
    }

    public Vehiculo fabricarVehiculo(){
        Vehiculo nuevoVehiculo = new Vehiculo();
        System.out.println("La clase Fabrica esta ejecutando el metodo fabricarVehiculo");
        return nuevoVehiculo;
    }

}
```

```
Waffles@Ulises-MacBook-Air: [~/Documents/UNAM/3° Semestre/P00/P4_P00/code/P4 (master)]$ cat Vehiculo.java
import java.util.ArrayList;

public class Vehiculo{

    private String marca;
    private String modelo;
    private ArrayList<Llanta> llantas;
    private Motor motor;

    public String getMarca(){
        return this.marca;
    }

    public void setMarca(String marca){
        this.marca = marca;
    }

    public String getModelo(){
        return this.modelo;
    }

    public void setModelo(String modelo){
        this.modelo = modelo;
    }

    public ArrayList<Llanta> getLlantas(){
        return this.llantas;
    }

    public void setLlantas(ArrayList<Llanta> llantas){
        this.llantas = llantas;
    }

    public Motor getMotor(){
        return this.motor;
    }

    public void setMotor(Motor motor){
        this.motor = motor;
    }

    public void acelerar(){
        System.out.println("La clase Vehiculo esta ejecutando el metodo acelerar");
    }

    public void frenar(){
        System.out.println("La clase Vehiculo esta ejecutando el metodo frenar");
    }

}
```

```
Waffles@Ulises-MacBook-Air: [~/Documents/UNAM/3° Semestre/P00/P4_P00/code/P4 (master)]$ cat Llanta.java
public class Llanta{

    private String material;
    private int diametro;
    private int presion;

    public String getMaterial(){
        return this.material;
    }

    public void setMaterial(String material){
        this.material = material;
    }

    public int getDiametro(){
        return this.diametro;
    }

    public void setDiametro(int diametro){
        this.diametro = diametro;
    }

    public int getPresion(){
        return this.presion;
    }

    public void setPresion(int presion){
        this.presion = presion;
    }

    public void aumentarPresion(int cantidad){
        System.out.println("La clase Llanta esta ejecutando el metodo aumentarPresion");
    }

    public void disminuirPresion(int cantida){
        System.out.println("La clase Llanta esta ejecutando el metodo disminuirPresion");
    }

}
```

```
Waffles@Ulises-MacBook-Air: [~/Documents/UNAM/3° Semestre/P00/P4_P00/code/P4 (master)]$ cat Motor.java
public class Motor{

    private String marca;
    private String modelo;
    private int potencia;

    public String getMarca(){
        return this.marca;
    }

    public void setMarca(String marca){
        this.marca = marca;
    }

    public String getModelo(){
        return this.modelo;
    }

    public void setModelo(String modelo){
        this.modelo = modelo;
    }

    public int getPotencia(){
        return this.potencia;
    }

    public void setPotencia(int potencia){
        this.potencia = potencia;
    }

}
```

```
Waffles@Ulises-MacBook-Air: [~/Documents/UNAM/3° Semestre/P00/P4_P00/code/P4 (master)]$ cat Automovil.java
import java.util.ArrayList;

public class Automovil extends Vehiculo{

    private String color;
    private Quemacocos accesorio;

    public String getColor(){
        return this.color;
    }

    public void setColor(String color){
        this.color = color;
    }

    public Quemacocos getAccesorio(){
        return this.accesorio;
    }

    public void setAccesorio(Quemacocos accesorio){
        this.accesorio = accesorio;
    }

}
```

```
Waffles@Ulises-MacBook-Air: [~/Documents/UNAM/3° Semestre/P00/P4_P00/code/P4 (master)]$ cat Motocicleta.java
import java.util.ArrayList;
```

```
public class Motocicleta extends Vehiculo{

    private String tipo;
    private Silenciador accesorio;

    public String getTipo(){
        return this.tipo;
    }

    public void setTipo(String tipo){
        this.tipo = tipo;
    }

    public Silenciador getAccesorio(){
        return this.accesorio;
    }

    public void setAccesorio(Silenciador accesorio){
        this.accesorio = accesorio;
    }

}
```

```
Waffles@Ulises-MacBook-Air: [~/Documents/UNAM/3° Semestre/P00/P4_P00/code/P4 (master)]$ cat Silenciador.java
public class Silenciador{
```

```
    private String marca;

    public String getMarca(){
        return this.marca;
    }

    public void setMarca(String marca){
        this.marca = marca;
    }

    public void suprimirRuido(){
        System.out.println("La clase Silenciador esta ejecutando el metodo suprimirRuido");
    }

}
```

```
Waffles@Ulises-MacBook-Air: [~/Documents/UNAM/3° Semestre/P00/P4_P00/code/P4 (master)]$ cat Quemacocos.java
public class Quemacocos{
```

```
    private boolean electrico;

    public boolean getElectrico(){
        return this.electrico;
    }

    public void setElectrico(boolean electrico){
        this.electrico = electrico;
    }

    public void abrir(){
        System.out.println("La clase Quemacocos esta ejecutando el metodo abrir");
    }

    public void cerrar(){
        System.out.println("La clase Quemacocos esta ejecutando el metodo cerrar");
    }

}
```


Compilación

```
Waffles@Ulises-MacBook-Air: [~/Documents/UNAM/3º Semestre/P00/P4_P00/code/P4 (master)]$ ls
total 72
drwxr-xr-x  11 Waffles  staff   374B Sep 10 11:14 ./
-rwxrwxrwx   1 Waffles  staff   531B Sep 10 11:13 Fabrica.java*
-rwxrwxrwx   1 Waffles  staff   870B Sep 10 11:06 Vehiculo.java*
-rwxrwxrwx   1 Waffles  staff   288B Sep 10 10:59 Silenciador.java*
-rwxrwxrwx   1 Waffles  staff    32B Sep 10 10:58 Distribuidora.java*
-rwxrwxrwx   1 Waffles  staff   391B Sep 10 10:53 Automovil.java*
-rwxrwxrwx   1 Waffles  staff   389B Sep 10 10:51 Motocicleta.java*
drwxrwxrwx  13 Waffles  staff   442B Sep 10 10:49 ./
-rwxrwxrwx   1 Waffles  staff   461B Sep  6 18:41 Motor.java*
-rwxrwxrwx   1 Waffles  staff   742B Sep  6 18:41 Llanta.java*
-rwxrwxrwx   1 Waffles  staff   407B Sep  6 18:28 Quemacocos.java*
```

```
Waffles@Ulises-MacBook-Air: [~/Documents/UNAM/3º Semestre/P00/P4_P00/code/P4 (master)]$ javac Fabrica.java
Waffles@Ulises-MacBook-Air: [~/Documents/UNAM/3º Semestre/P00/P4_P00/code/P4 (master)]$ ls
total 104
drwxr-xr-x  15 Waffles  staff   510B Sep 10 11:15 ./
-rw-r--r--   1 Waffles  staff   838B Sep 10 11:15 Fabrica.class
-rw-r--r--   1 Waffles  staff   1.1K Sep 10 11:15 Llanta.class
-rw-r--r--   1 Waffles  staff   698B Sep 10 11:15 Motor.class
-rw-r--r--   1 Waffles  staff   1.4K Sep 10 11:15 Vehiculo.class
-rwxrwxrwx   1 Waffles  staff   531B Sep 10 11:13 Fabrica.java*
-rwxrwxrwx   1 Waffles  staff   870B Sep 10 11:06 Vehiculo.java*
-rwxrwxrwx   1 Waffles  staff   288B Sep 10 10:59 Silenciador.java*
-rwxrwxrwx   1 Waffles  staff    32B Sep 10 10:58 Distribuidora.java*
-rwxrwxrwx   1 Waffles  staff   391B Sep 10 10:53 Automovil.java*
-rwxrwxrwx   1 Waffles  staff   389B Sep 10 10:51 Motocicleta.java*
drwxrwxrwx  13 Waffles  staff   442B Sep 10 10:49 ./
-rwxrwxrwx   1 Waffles  staff   461B Sep  6 18:41 Motor.java*
-rwxrwxrwx   1 Waffles  staff   742B Sep  6 18:41 Llanta.java*
-rwxrwxrwx   1 Waffles  staff   407B Sep  6 18:28 Quemacocos.java*
```

```
Waffles@Ulises-MacBook-Air: [~/Documents/UNAM/3º Semestre/P00/P4_P00/code/P4 (master)]$ javac Automovil.java
Waffles@Ulises-MacBook-Air: [~/Documents/UNAM/3º Semestre/P00/P4_P00/code/P4 (master)]$ ls
total 120
drwxr-xr-x  17 Waffles  staff   578B Sep 10 11:16 ./
-rw-r--r--   1 Waffles  staff   592B Sep 10 11:16 Automovil.class
-rw-r--r--   1 Waffles  staff   727B Sep 10 11:16 Quemacocos.class
-rw-r--r--   1 Waffles  staff   838B Sep 10 11:15 Fabrica.class
-rw-r--r--   1 Waffles  staff   1.1K Sep 10 11:15 Llanta.class
-rw-r--r--   1 Waffles  staff   698B Sep 10 11:15 Motor.class
-rw-r--r--   1 Waffles  staff   1.4K Sep 10 11:15 Vehiculo.class
-rwxrwxrwx   1 Waffles  staff   531B Sep 10 11:13 Fabrica.java*
-rwxrwxrwx   1 Waffles  staff   870B Sep 10 11:06 Vehiculo.java*
-rwxrwxrwx   1 Waffles  staff   288B Sep 10 10:59 Silenciador.java*
-rwxrwxrwx   1 Waffles  staff    32B Sep 10 10:58 Distribuidora.java*
-rwxrwxrwx   1 Waffles  staff   391B Sep 10 10:53 Automovil.java*
-rwxrwxrwx   1 Waffles  staff   389B Sep 10 10:51 Motocicleta.java*
drwxrwxrwx  13 Waffles  staff   442B Sep 10 10:49 ./
-rwxrwxrwx   1 Waffles  staff   461B Sep  6 18:41 Motor.java*
-rwxrwxrwx   1 Waffles  staff   742B Sep  6 18:41 Llanta.java*
-rwxrwxrwx   1 Waffles  staff   407B Sep  6 18:28 Quemacocos.java*
```

```

[Waffles@Ulises-MacBook-Air:~/Documents/UNAM/3º Semestre/P00/P4_P00/code/P4 (master)]$ javac Motocicleta.java
[Waffles@Ulises-MacBook-Air:~/Documents/UNAM/3º Semestre/P00/P4_P00/code/P4 (master)]$ ls
total 136
drwxr-xr-x  19 Waffles  staff   646B Sep 10 11:16 ./
-rw-r--r--   1 Waffles  staff   596B Sep 10 11:16 Motocicleta.class
-rw-r--r--   1 Waffles  staff   643B Sep 10 11:16 Silenciador.class
-rw-r--r--   1 Waffles  staff   592B Sep 10 11:16 Automovil.class
-rw-r--r--   1 Waffles  staff   727B Sep 10 11:16 Quemacocos.class
-rw-r--r--   1 Waffles  staff   838B Sep 10 11:15 Fabrica.class
-rw-r--r--   1 Waffles  staff   1.1K Sep 10 11:15 Llanta.class
-rw-r--r--   1 Waffles  staff   698B Sep 10 11:15 Motor.class
-rw-r--r--   1 Waffles  staff   1.4K Sep 10 11:15 Vehiculo.class
-rwxrwxrwx   1 Waffles  staff   531B Sep 10 11:13 Fabrica.java*
-rwxrwxrwx   1 Waffles  staff   870B Sep 10 11:06 Vehiculo.java*
-rwxrwxrwx   1 Waffles  staff   288B Sep 10 10:59 Silenciador.java*
-rwxrwxrwx   1 Waffles  staff    32B Sep 10 10:58 Distribuidora.java*
-rwxrwxrwx   1 Waffles  staff   391B Sep 10 10:53 Automovil.java*
-rwxrwxrwx   1 Waffles  staff   389B Sep 10 10:51 Motocicleta.java*
drwxrwxrwx  13 Waffles  staff   442B Sep 10 10:49 ./
-rwxrwxrwx   1 Waffles  staff   461B Sep  6 18:41 Motor.java*
-rwxrwxrwx   1 Waffles  staff   742B Sep  6 18:41 Llanta.java*
-rwxrwxrwx   1 Waffles  staff   407B Sep  6 18:28 Quemacocos.java*

```

```

[Waffles@Ulises-MacBook-Air:~/Documents/UNAM/3º Semestre/P00/P4_P00/code/P4 (master)]$ javac Distribuidora.java
[Waffles@Ulises-MacBook-Air:~/Documents/UNAM/3º Semestre/P00/P4_P00/code/P4 (master)]$ ls
total 144
drwxr-xr-x  20 Waffles  staff   680B Sep 10 11:17 ./
-rw-r--r--   1 Waffles  staff   200B Sep 10 11:17 Distribuidora.class
-rw-r--r--   1 Waffles  staff   596B Sep 10 11:16 Motocicleta.class
-rw-r--r--   1 Waffles  staff   643B Sep 10 11:16 Silenciador.class
-rw-r--r--   1 Waffles  staff   592B Sep 10 11:16 Automovil.class
-rw-r--r--   1 Waffles  staff   727B Sep 10 11:16 Quemacocos.class
-rw-r--r--   1 Waffles  staff   838B Sep 10 11:15 Fabrica.class
-rw-r--r--   1 Waffles  staff   1.1K Sep 10 11:15 Llanta.class
-rw-r--r--   1 Waffles  staff   698B Sep 10 11:15 Motor.class
-rw-r--r--   1 Waffles  staff   1.4K Sep 10 11:15 Vehiculo.class
-rwxrwxrwx   1 Waffles  staff   531B Sep 10 11:13 Fabrica.java*
-rwxrwxrwx   1 Waffles  staff   870B Sep 10 11:06 Vehiculo.java*
-rwxrwxrwx   1 Waffles  staff   288B Sep 10 10:59 Silenciador.java*
-rwxrwxrwx   1 Waffles  staff    32B Sep 10 10:58 Distribuidora.java*
-rwxrwxrwx   1 Waffles  staff   391B Sep 10 10:53 Automovil.java*
-rwxrwxrwx   1 Waffles  staff   389B Sep 10 10:51 Motocicleta.java*
drwxrwxrwx  13 Waffles  staff   442B Sep 10 10:49 ./
-rwxrwxrwx   1 Waffles  staff   461B Sep  6 18:41 Motor.java*
-rwxrwxrwx   1 Waffles  staff   742B Sep  6 18:41 Llanta.java*
-rwxrwxrwx   1 Waffles  staff   407B Sep  6 18:28 Quemacocos.java*

```

Ejercicio 2

Código


```
public class Fabrica//nombre de la clase
{
    private String nombre;
    private String direccion;

    public void setNombre(String nombre)
    {
        this.nombre = nombre;
    }
    public String getNombre()
    {
        return this.nombre;
    }
    public void setDireccion(String direccion)
    {
        this.direccion = direccion;
    }
    public String getDireccion()
    {
        return this.direccion;
    }
    public Fabrica()//constructor vacio
    {
        this.nombre = "";
        this.direccion = "";
    }
    public Fabrica(String nombre, String direccion)//constructor sobrecargado
    {
        this.nombre=nombre;
        this.direccion=direccion;
    }
    public Vehiculo fabricarVehiculo()
    {
        Vehiculo nuevoVehiculo = new Vehiculo();
        System.out.println("La clase Fabrica esta ejecutando el método fabricarVehiculo");
        return nuevoVehiculo;
    }
}
```

```
import java.util.ArrayList;
public class Vehiculo
{
    protected String marca;
    protected String modelo;
    private ArrayList<Llanta> llantas;
    private Motor motor;

    public void setMarca(String marca)//uso de set
    {
        this.marca = marca;
    }
    public String getMarca()//uso de get
    {
        return this.marca;
    }
    public void setModelo(String modelo)
    {
        this.modelo = modelo;
    }
    public String modelo()
    {
        return this.modelo;
    }
    public void acelerar()
    {
        System.out.println("La clase Vehiculo esta ejecutando el método acelerar");//método acelerar
    }
    public void frenar()
    {
        System.out.println("La clase Vehiculo esta ejecutando el método frenar");//método frenar
    }
    public Vehiculo()//Constructor vacío
    {
        this.marca = "";
        this.modelo = "";
        this.llantas = new ArrayList<Llanta>(2);
    }
    public Vehiculo(String marca, String modelo)//constructor sobrecargado
    {
        this.marca =marca;
        this.modelo =modelo;
    }
}
```

```
private String material;//privado
private int diametro;
private int presion;

public void setMaterial(String material)//uso de set
{
    this.material = material;
}
public String getMaterial()//uso de get
{
    return this.material;
}
public void setDiametro(int diametro)
{
    this.diametro = diametro;
}
public int diametro()
{
    return this.diametro;
}
public void setPresion(int presion)
{
    this.presion = presion;
}
public int getDiametro()
{
    return this.diametro;
}
public void aumentarPresion(int cantidad)//método aumentarPresion
{
    System.out.println("La clase Llanta esta ejecutando el método aumentarPresion");
}
public void disminuirPresion(int cantidad)//método disminuirPresion
{
    System.out.println("La clase Llanta esta ejecutando el método disminuirPresion");
}
public Llanta()//Constructor vacío
{
    this.material="";
    this.diametro=0;
    this.presion=0;
}
public Llanta(String material, int diametro, int presion)//Constructor sobrecargado
{
    this.material=material;
    this.diametro=diametro;
    this.presion=presion;
}
```

```

public class Motor//nombre de la clase
{
    private String marca;//privados
    private String modelo;
    private int potencia;

    public void setMarca(String marca)//uso de set para poder asignar
                                   //un valor al atributo privado marca
    {
        this.marca = marca;
    }
    public String getMarca()//uso de get para obtener el valor
                                   //asignado a marca
    {
        return this.marca;
    }
    public void setModelo(String modelo)//uso de set para modelo
    {
        this.modelo = modelo;
    }
    public String getModelo()//uso de get para modelo
    {
        return this.modelo;
    }
    public void setPotencia(int potencia)//uso de set para potencia
    {
        this.potencia = potencia;
    }
    public int getPotencia()//utilización de get para potencia
    {
        return this.potencia;
    }
    public Motor()//constructor vacío
    {
        this.marca = "";
        this.modelo = "";
        this.potencia=0;
    }
    public Motor(String marca, String modelo, int potencia)//constructor sobrecargado
    {
        this.marca=marca;
        this.modelo=modelo;
        this.potencia=potencia;
    }
}

```

```

import java.util.ArrayList;

public class Automovil extends Vehiculo//herencia de la clase Vehiculo
{
    private String color;//privado

    public void setColor(String color)//uso de set
    {
        this.color = color;
    }
    public String getColor()//uso de get
    {
        return this.color;
    }

    public Automovil()//constructor vacío
    {
        this.color="";
        super.marca="";//uso de los atributos de la clase padre
        super.modelo="";
    }
    public Automovil(String color, String marca, String modelo)//constructor sobrecargado
    {
        super.marca=marca;
        super.modelo=modelo;
        this.color=color;
    }
}

```



```

public class Quemacocos//nombre de la clase
{
    private boolean electrico;//atributo privado tipo booleano

    public void setElectrico(boolean electrico)//uso de set
    {
        this.electrico = electrico;
    }
    public boolean getElectrico()//uso de get
    {
        return this.electrico;
    }
    public void abrir()//método abrir
    {
        System.out.println("La clase Quemacocos esta ejecutando el método abrir");
    }
    public void cerrar()//método cerrar
    {
        System.out.println("La clase Quemacocos esta ejecutando el método cerrar");
    }
    public Quemacocos()//constructor vacío
    {
        this.electrico=false;
    }
    public Quemacocos(boolean electrico)//constructor sobrecargado
    {
        this.electrico=electrico;
    }
}

```

```

import java.util.ArrayList;
public class Motocicleta extends Vehiculo//herencia de la clase Vehiculo
{
    private String tipo;//atributo privado

    public void setTipo(String tipo)//uso de set
    {
        this.tipo = tipo;
    }
    public String getTipo()//uso de get
    {
        return this.tipo;
    }
    public Motocicleta()//constructor vacío
    {
        this.tipo="";
        super.marca="";
        super.modelo="";
    }
    public Motocicleta(String tipo, String marca, String modelo)//Constructor sobrecargado
    {
        this.tipo=tipo;
        super.marca=marca;
        super.modelo=modelo;
    }
}

```

```
public class Silenciador//nombre de la clase
{
    private String marca;//atributo privado

    public void setMarca(String marca)//uso de set
    {
        this.marca = marca;
    }
    public String getMarca()//uso de get
    {
        return this.marca;
    }
    public void suprimirRuido()//método suprimirRuido
    {
        System.out.println("La clase Silenciador esta ejecutando el método suprimirRuido");
    }
    public Silenciador()//constructor vacío
    {
        this.marca="";
    }
    public Silenciador(String marca)//constructor sobrecargado
    {
        this.marca=marca;
    }
}
```

Compilación

```
Sarahi MH@LAPTOP-0BM4RB5E MINGW64 ~/Documents/Codigos_POO/Practica4/Ejercicio 2
$ vi Fabrica.java

Sarahi MH@LAPTOP-0BM4RB5E MINGW64 ~/Documents/Codigos_POO/Practica4/Ejercicio 2
$ javac Fabrica.java

Sarahi MH@LAPTOP-0BM4RB5E MINGW64 ~/Documents/Codigos_POO/Practica4/Ejercicio 2
$ vi Vehiculo.java

Sarahi MH@LAPTOP-0BM4RB5E MINGW64 ~/Documents/Codigos_POO/Practica4/Ejercicio 2
$ javac Vehiculo.java

Sarahi MH@LAPTOP-0BM4RB5E MINGW64 ~/Documents/Codigos_POO/Practica4/Ejercicio 2
$ vi Llanta.java

Sarahi MH@LAPTOP-0BM4RB5E MINGW64 ~/Documents/Codigos_POO/Practica4/Ejercicio 2
$ javac Llanta.java
```

```
Sarahi MH@LAPTOP-OBM4RB5E MINGW64 ~/Documents/Codigos_P00/Practica4/Ejercicio 2
$ vi Motor.java

Sarahi MH@LAPTOP-OBM4RB5E MINGW64 ~/Documents/Codigos_P00/Practica4/Ejercicio 2
$ javac Motor.java

Sarahi MH@LAPTOP-OBM4RB5E MINGW64 ~/Documents/Codigos_P00/Practica4/Ejercicio 2
$ vi Motor.java

Sarahi MH@LAPTOP-OBM4RB5E MINGW64 ~/Documents/Codigos_P00/Practica4/Ejercicio 2
$ javac Motor.java

Sarahi MH@LAPTOP-OBM4RB5E MINGW64 ~/Documents/Codigos_P00/Practica4/Ejercicio 2
$ vi Automovil.java

Sarahi MH@LAPTOP-OBM4RB5E MINGW64 ~/Documents/Codigos_P00/Practica4/Ejercicio 2
$ javac Automovil.java

Sarahi MH@LAPTOP-OBM4RB5E MINGW64 ~/Documents/Codigos_P00/Practica4/Ejercicio 2
$ vi Quemacocos.java

Sarahi MH@LAPTOP-OBM4RB5E MINGW64 ~/Documents/Codigos_P00/Practica4/Ejercicio 2
$ javac Quemacocos.java

Sarahi MH@LAPTOP-OBM4RB5E MINGW64 ~/Documents/Codigos_P00/Practica4/Ejercicio 2
$ vi Motocicleta.java

Sarahi MH@LAPTOP-OBM4RB5E MINGW64 ~/Documents/Codigos_P00/Practica4/Ejercicio 2
$ javac Motocicleta.java

Sarahi MH@LAPTOP-OBM4RB5E MINGW64 ~/Documents/Codigos_P00/Practica4/Ejercicio 2
$ vi Silenciador.java

Sarahi MH@LAPTOP-OBM4RB5E MINGW64 ~/Documents/Codigos_P00/Practica4/Ejercicio 2
$ javac Silenciador.java
```

```
Sarahi MH@LAPTOP-OBM4RB5E MINGW64 ~/Documents/Codigos_P00/Practica4/Ejercicio 2
$ ls
Automovil.class  Distribuidora.class  Fabrica.class  Llanta.class  Motocicleta.class  Motor.class  Quemacocos.class  Silenciador.class  Vehiculo.class
Automovil.java  Distribuidora.java  Fabrica.java  Llanta.java  Motocicleta.java  Motor.java  Quemacocos.java  Silenciador.java  Vehiculo.java
```

Ejercicio 3

Código

```
Waffles@Ulises-MacBook-Air: [~/Documents/UNAM/3º Semestre/P00/P4_P00/code/Ejercicio 3 (master)]$ cat Distribuidora.java
import java.util.Random;
import java.util.ArrayList;

public class Distribuidora{
    private ArrayList vehiculos = new ArrayList(); // ArrayList de vehiculos, ya sean motocicletas o automóviles
    private Fabrica fabrica = new Fabrica("Fabrica de autos", "Av. del IMAN 704"); // Se crea una instancia de la clase fábrica con un nombre y dirección
    Random rnd = new Random(); // Objeto rnd de la clase Random para poder generar aleatoriamente el número de autos y motocicletas
    public static void main(String[] args){
        Distribuidora miDistribuidora = new Distribuidora();
        int numero_de_autos = ((miDistribuidora.rnd.nextInt(5))+1); // Número de autos entre 1 y 5
        int numero_de_motos = 5 - numero_de_autos; // Número de motocicletas necesario para completar 5 vehiculos
        System.out.println(numero_de_autos+" autos y "+numero_de_motos+" motos serán fabricados.");
        for(int i = 1; i<= numero_de_autos; i++){
            miDistribuidora.vehiculos.add(miDistribuidora.fabrica.fabricarVehiculo(true)); // Se crean los autos llamando al método fabricarVehiculo del objeto fábrica con argumento true,
            // para indicar que se solicita un auto
        }
        for(int i = 1; i<= numero_de_motos; i++){
            miDistribuidora.vehiculos.add(miDistribuidora.fabrica.fabricarVehiculo(false)); // Cuando se llama al método fabricarVehiculo con argumento false se indica que se requieren motocic
        }
        for(int i = 0; i<miDistribuidora.vehiculos.size(); i++){
            System.out.println(miDistribuidora.vehiculos.get(i).toString()); // Se imprimen las características del vehiculo mediante el método toString sobrescrito
        }
        for(int i = 0; i<5; i++){
            System.out.println("Probando vehiculo número "+(i+1));
            String tipo_de_vehiculo = miDistribuidora.vehiculos.get(i).getClass().getName();
            if(tipo_de_vehiculo == "Motocicleta"){
                Motocicleta moto_en_prueba = (Motocicleta) miDistribuidora.vehiculos.get(i);
                Silenciador accesorio = moto_en_prueba.getAccesorio();
                if(accesorio != null){ // En caso de que el vehiculo cuente con accesorio, se procede a probarlo
                    accesorio.suprimirRuido();
                }
            } else {
                Automovil auto_en_prueba = (Automovil) miDistribuidora.vehiculos.get(i);
                Quemacocos accesorio = auto_en_prueba.getAccesorio();
                if(accesorio != null){ // En caso de que el vehiculo cuente con accesorio, se procede a probarlo
                    accesorio.abrir(); // Como el accesorio es un quemacocos, se procede a ejecutar sus métodos
                    accesorio.cerrar();
                }
            }
        }
    }
}
```



```
Waffles@Ulises-MacBook-Air: [~/Documents/UNAM/3º Semestre/P00/P4_P00/code/Ejercicio 3 (master)]$ cat Fabrica.java
import java.util.Scanner;
import java.util.ArrayList;

public class Fabrica
{
    private String nombre;
    private String direccion;
    Scanner sc= new Scanner(System.in);

    public void setNombre(String nombre)
    {
        this.nombre = nombre;
    }
    public String getNombre()
    {
        return this.nombre;
    }
    public void setDireccion(String direccion)
    {
        this.direccion = direccion;
    }
    public String getDireccion()
    {
        return this.direccion;
    }
    public Fabrica()
    {
        this.nombre = "";
        this.direccion = "";
    }

    public Fabrica(String nombre, String direccion)
    {
        this.nombre = nombre;
        this.direccion = direccion;
    }
    public Vehiculo fabricarVehiculo(boolean automovil)
    {
        String marca="";
        String color="";
        String aosdjoas="";
        int presion;
        if(automovil){
            Automovil nuevo_auto = new Automovil();
            System.out.println("Introduzca marca de automóvil: ");
            marca = sc.next();
            nuevo_auto.setMarca(marca);
            System.out.println("Introduzca modelo de automóvil: ");
            aosdjoas= sc.next();
            nuevo_auto.setModelo(aosdjoas);
            System.out.println("Introduzca color de automóvil: ");
            color = sc.next();
            nuevo_auto.setColor(color);
        }
    }
}
```

```

        int quemacocos;;
        do {
            System.out.println("¿Desea que el automóvil tenga quemacocos? (Introduzca 1 si desea quemacocos o 0 si no lo desea)");
            quemacocos = sc.nextInt();
            if(quemacocos != 1 && quemacocos != 0)
                System.out.println("Opción incorrecta. Vuelva a intentarlo.");
        } while(quemacocos != 1 && quemacocos != 0);
        if(quemacocos == 1){
            do{
                System.out.println("Si desea que el quemacocos sea electrico, introduzca 1, en caso contrario, introduzca 0");
                quemacocos = sc.nextInt();
                if(quemacocos != 1 && quemacocos != 0)
                    System.out.println("Opción incorrecta. Vuelva a intentarlo.");
            }while(quemacocos != 1 && quemacocos != 0);
            if(quemacocos == 1){
                Quemacocos accesorio = new Quemacocos(true);
                nuevo_auto.setAccesorio(accesorio);
            } else {
                Quemacocos accesorio = new Quemacocos(false);
                nuevo_auto.setAccesorio(accesorio);
            }
        }
        do{
            System.out.println("Por favor ingrese la presión que desea en sus llantas: ");
            presion= sc.nextInt();
            if(presion>0 && presion<=10){
                ArrayList<Llanta> llantas = new ArrayList<Llanta>();
                for(int i=0; i<4; i++){
                    Llanta llanta = new Llanta();
                    llanta.setPresion(presion);
                    llantas.add(llanta);
                }
                nuevo_auto.setLlantas(llantas);
                System.out.println("La presión actual de sus llantas es: "+presion);
            }else{
                System.out.println("La presión que ingresó no es adecuada, por favor intente de nuevo");
            }
        }while(presion<1 || presion>10);
        return nuevo_auto;
    } else {
        Motocicleta nueva_moto = new Motocicleta();
        System.out.println("Introduzca marca de motocicleta: ");
        marca = sc.next();
        nueva_moto.setMarca(marca);
        System.out.println("Introduzca modelo de motocicleta: ");
        String modelo = sc.next();
        nueva_moto.setModelo(modelo);
        System.out.println("Introduzca tipo de motocicleta: ");
        String tipo="";
        tipo = sc.next();
        nueva_moto.setTipo(tipo);
        int silenciador;
    }
}

```

```

        do {
            System.out.println("¿Desea que la moto tenga silenciador? (Introduzca 1 si desea silenciado
r o 0 si no lo desea)");
            silenciador = sc.nextInt();
            if(silenciador != 1 && silenciador != 0)
                System.out.println("Opción incorrecta. Vuelva a intentarlo.");
        } while(silenciador != 1 && silenciador != 0);
        if(silenciador == 1){
            System.out.println("Introduzca marca de silenciador: ");
            marca = sc.next();
            Silenciador accesorio = new Silenciador(marca);
            nueva_moto.setAccesorio(accesorio);
        }
        do{
            System.out.println("Por favor ingrese la presión que desea en sus llantas: ");
            presion= sc.nextInt();
            if(presion>0 && presion<=10){
                ArrayList<Llanta> llantas = new ArrayList<Llanta>();
                for(int i=0; i<2; i++){
                    Llanta llanta = new Llanta();
                    llanta.setPresion(presion);
                    llantas.add(llanta);
                }
                nueva_moto.setLlantas(llantas);
                System.out.println("La presión actual de sus llantas es: "+presion);
            }else{
                System.out.println("La presión que ingresó no es adecuada, por favor intente de nuevo");
            }
        }while(presion<1 || presion>10);
        return nueva_moto;
    }
}
}

```

Waffles@Ulises-MacBook-Air: [~/Documents/UNAM/3° Semestre/P00/P4_P00/code/Ejercicio 3 (master)]\$ cat Vehiculo.java

```
import java.util.ArrayList;
import java.util.Scanner;

public class Vehiculo
{
    protected String marca;
    protected String modelo;
    private ArrayList<Llanta> llantas;
    private Motor motor;
    Scanner scanner = new Scanner(System.in);

    public void setMarca(String marca)
    {
        this.marca = marca;
    }
    public String getMarca()
    {
        return this.marca;
    }
    public void setModelo(String modelo)
    {
        this.modelo = modelo;
    }
    public String modelo()
    {
        return this.modelo;
    }
    public void setLlantas(ArrayList<Llanta> llantas){
        this.llantas = llantas;
    }
    public void acelerar()
    {
        System.out.println("La clase Vehiculo esta ejecutando el método acelerar");
    }
    public void frenar()
    {
        System.out.println("La clase Vehiculo esta ejecutando el método frenar");
    }
    public Vehiculo()
    {
        this.marca = "";
        this.modelo = "";
        this.llantas = new ArrayList<Llanta>();
    }
    public Vehiculo(String marca, String modelo)
    {
        this.marca = marca;
        this.modelo = modelo;
    }

    public String toString(){
        if(llantas.size() == 2)
            return "Motocicleta marca "+this.marca+" y modelo "+this.modelo;
        else
            return "Automóvil marca "+this.marca+" y modelo "+this.modelo;
        }
    }
}
```



```
Waffles@Ulises-MacBook-Air:[~/Documents/UNAM/3° Semestre/P00/P4_P00/code/Ejercicio 3 (master)]$ cat Automovil.java
import java.util.Scanner;
import java.util.ArrayList;

public class Automovil extends Vehiculo//palabra reservada para poder denotar
                                         //la herencia de la clase Vehiculo a Automovil y así poder
                                         //utilizar los atributos y operaciones de la Clase Padre
{
    private String color;
    private Quemacocos accesorio;

    public void setColor(String color)//utilización del set para asignar un valor al atributo privado color
    {
        this.color = color;
    }
    public String getColor()//utilización de get para obtener el color
    {
        return this.color;
    }

    public void setAccesorio(Quemacocos quemacocos){
        this.accesorio = quemacocos;
    }

    public Quemacocos getAccesorio(){
        return this.accesorio;
    }

    public Automovil()//constructor vacío que inicializa los valores para que no sean null
    {
        super.marca = "";//indica que el atributo marca se heredó de Vehiculo
        super.modelo = "";//el atributo modelo se heredó de Vehiculo
        this.color="";
    }
    public Automovil(String color, String marca, String modelo)//Constructor sobrecargado
    {
        super.marca = marca;
        super.modelo = modelo;
        this.color = color;
    }
}
```

```
Waffles@Ulises-MacBook-Air: [~/Documents/UNAM/3° Semestre/P00/P4_P00/code/Ejercicio 3 (master)]$ cat Motocicleta.java
import java.util.ArrayList;
import java.util.Scanner;

public class Motocicleta extends Vehiculo
{
    private String tipo;
    private Silenciador accesorio;
    public void setTipo(String tipo)
    {
        this.tipo = tipo;
    }
    public String getTipo()
    {
        return this.tipo;
    }

    public void setAccesorio(Silenciador accesorio){
        this.accesorio = accesorio;
    }

    public Silenciador getAccesorio(){
        return this.accesorio;
    }

    public Motocicleta()
    {
        super.marca = "";
        super.modelo = "";
        this.tipo="";
    }
    public Motocicleta(String tipo, String marca, String modelo)
    {
        this.tipo = tipo;
        super.marca = marca;
        super.modelo = modelo;
    }
}
```

Compilación y ejecución

```
Waffles@Ulises-MacBook-Air: [~/Documents/UNAM/3º Semestre/P00/P4_P00/code/Ejercicio 3 (master)]$ javac Fabrica.java
Waffles@Ulises-MacBook-Air: [~/Documents/UNAM/3º Semestre/P00/P4_P00/code/Ejercicio 3 (master)]$ javac Distribuidora.java
Note: Distribuidora.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Waffles@Ulises-MacBook-Air: [~/Documents/UNAM/3º Semestre/P00/P4_P00/code/Ejercicio 3 (master)]$ java Distribuidora
1 autos y 4 motos serán fabricados.
Introduzca marca de automóvil:
BMW
Introduzca modelo de automóvil:
i3
Introduzca color de automóvil:
Azul
¿Desea que el automóvil tenga quemacocos? (Introduzca 1 si desea quemacocos o 0 si no lo desea)
0
Por favor ingrese la presión que desea en sus llantas:
10
La presión actual de sus llantas es: 10
Introduzca marca de motocicleta:
Suzuki
Introduzca modelo de motocicleta:
GN125
Introduzca tipo de motocicleta:
Ciudad
¿Desea que la moto tenga silenciador? (Introduzca 1 si desea silenciador o 0 si no lo desea)
1
Introduzca marca de silenciador:
Silentpro
Por favor ingrese la presión que desea en sus llantas:
7
La presión actual de sus llantas es: 7
Introduzca marca de motocicleta:
Yamaha
Introduzca modelo de motocicleta:
FZ16
Introduzca tipo de motocicleta:
Sport
¿Desea que la moto tenga silenciador? (Introduzca 1 si desea silenciador o 0 si no lo desea)
0
Por favor ingrese la presión que desea en sus llantas:
10
La presión actual de sus llantas es: 10
Introduzca marca de motocicleta:
Honda
Introduzca modelo de motocicleta:
CBR
Introduzca tipo de motocicleta:
Sport
¿Desea que la moto tenga silenciador? (Introduzca 1 si desea silenciador o 0 si no lo desea)
1
Introduzca marca de silenciador:
SportXtra
Por favor ingrese la presión que desea en sus llantas:
6
La presión actual de sus llantas es: 6
Introduzca marca de motocicleta:
```

```
Introduzca marca de motocicleta:
Italika
Introduzca modelo de motocicleta:
125N
Introduzca tipo de motocicleta:
Ciudad
¿Desea que la moto tenga silenciador? (Introduzca 1 si desea silenciador o 0 si no lo desea)
0
Por favor ingrese la presión que desea en sus llantas:
4
La presión actual de sus llantas es: 4
Automóvil marca BMW y modelo i3
Motocicleta marca Suzuki y modelo GN125
Motocicleta marca Yamaha y modelo FZ16
Motocicleta marca Honda y modelo CBR
Motocicleta marca Italika y modelo 125N
Probando vehículo número 1
Probando vehículo número 2
La clase Silenciador esta ejecutando el método suprimirRuido
Probando vehículo número 3
Probando vehículo número 4
La clase Silenciador esta ejecutando el método suprimirRuido
Probando vehículo número 5
```


Conclusiones

Ahuactzin Lopez Jorge Daniel

Esta práctica nos ayudó mucho a materializar un diagrama UML a un código en Java, ya que en este tipo de diagramas te dice que es lo que necesita el cliente, todas las especificaciones vienen implícitamente y el código debe correr a como esté estipulado, también nos ayudó con los constructores nulos para inicializar un contador

Landázuri Brambila Álvaro Ulises

En esta práctica se pudo implementar un diagrama de UML en clases de Java, que se comunican entre sí, utilizando modificadores de acceso y sobrecargando constructores. Sin embargo, el último ejercicio no pudo ser completado con éxito dentro del plazo indicado. A pesar de que se presentaron algunas dificultades en el transcurso de la práctica, mediante la consulta de la documentación de Java y la experimentación con código se pudo lograr un mayor dominio del lenguaje y, sobretodo, el paradigma orientado a objetos. El concepto de encapsulamiento fue comprendido a mucha mayor profundidad gracias a la práctica, pues al momento de experimentar diversos problemas derivados de su implementación se pudieron afianzar de manera intuitiva los conocimientos teóricos vistos dentro del aula. La práctica fue un excelente ejercicio para detectar nuestras debilidades en la codificación y poder trabajar en el mejoramiento de nuestras habilidades como programadores efectivos y proactivos.

Marcial Hernández Claudia Sarahi

Debido a los ejercicios propuestos de ésta práctica, comprendí cómo interpretar un diagrama UML para poder implementarlo en código, así como poder realizar la implementación de los conceptos vistos en clase, tal es el caso de los constructores, y cómo es la estructura de uno vacío, a diferencia de uno sobrecargado, el cual, se había visto el concepto en clase. Se utilizaron por primera vez los getters y setter, para poder acceder a los atributos privados de una clase, así mismo se utilizó la palabra reservada `super()`, el cual denotaba la herencia de una clase Padre, a una clase hija, para poder utilizar los atributos declarados de la Clase Padre. Tuve complicaciones a la hora de implementar algunos de los conceptos y palabras reservadas mencionadas anteriormente, pero después de leer y releer, lo teórico al final se compilaron los códigos y entendí mejor el cómo se usaban y en qué casos.

Medina Molina Fernando

Gracias a los ejercicios propuestos, pudimos comprender mejor e inclusive aprender desde el API de java más acerca de los constructores, además de cómo emplear las funciones de `this()` y `super()` para la creación de las clases con las que vamos a trabajar dentro de los códigos. Así mismo, gracias a los diagramas de UML proporcionados para la realización de los ejercicios, comprendimos su importancia dentro de la materia y cómo es que funciona, a lo largo de la práctica surgieron varias dudas que nos hicieron investigar por propia cuenta y enriquecer más el conocimiento que tenemos sobre el lenguaje.