Ciclo 2 Fundamentos de programación

Reto 2 – Grupo 18

Descripción del problema:

El concesionario **Aquí Tengo Tu Vehículo Nuevo** está implementando un sistema de pedidos para venta de vehículos al por mayor (venta de 2 o más vehículos distintos en la misma compra), el cual permite calcular el precio total de un pedido de varios vehículos dependiendo de las características que los vehículos cotizados.

Un pedido es hecho por un cliente en una fecha, y dicho pedido está compuesto por varios vehículos, los cuales pueden ser automóviles o motocicletas. Estos son de la marca y modelo que desee el cliente y dependiendo del vehículo, tendrá un precio base.

Los automóviles pueden tener transmisión manual o automática. Los automáticos tienen un incremento en el precio base del 5%. Si tiene vidrios eléctricos, el precio se incrementa en $ 400.000 y el Aire acondicionado incrementa en $3’000.000.

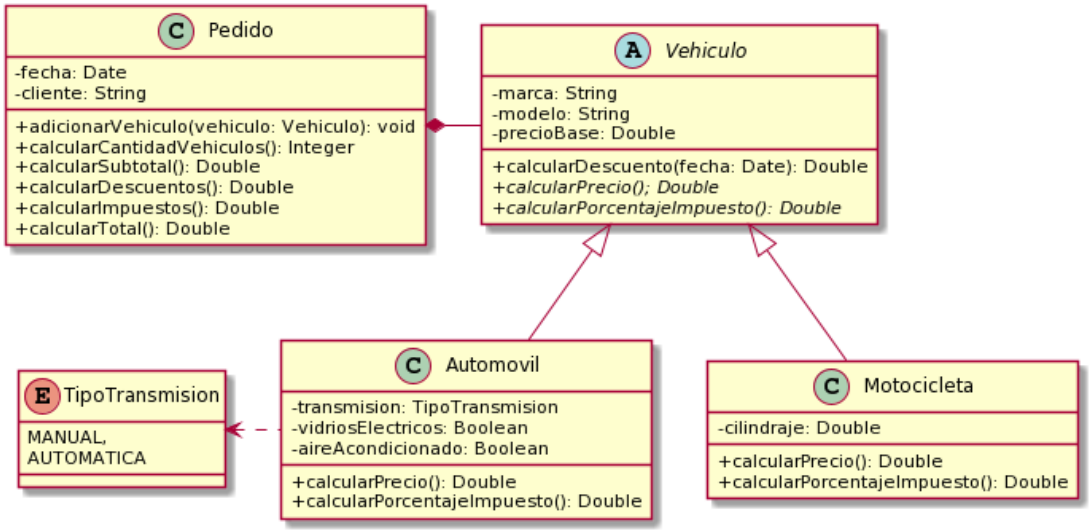
Las motocicletas que tienen un cilindraje entre 200cc y 350cc tienen un incremento en precio de $500.000, las de un cilindraje entre 351cc y 500cc tienen un incremento de $750.000, y las superiores a 500cc tienen un incremento en precio de $1’000.000.

Si el pedido se hizo en el mes de julio, los vehículos de la marca **BMW** tienen un descuento por el fabricante de un 15%. Y si en el mes de junio, **Suzuki** dio descuento de 10% en los automóviles y un 5% en motocicletas.

Por regulación nacional, los automóviles con precio superior a $60’000.000 deben pagar un impuesto IVA del 10%, igual que las motocicletas superiores a $7’500.000.

El sistema debe permitir agregar vehículos, calcular cuantos vehículos en total han sido pedidos, el subtotal (total acumulado de precios de los vehículos), el valor total de impuestos y el gran total (subtotal - descuentos + impuestos).

Para esto, te han contratado para que codifiques la solución al sistema del concesionario, y el equipo de diseño te ha entregado el siguiente modelo de clases: (link)



Ejemplo:

Prueba 1

```
var fecha = new SimpleDateFormat("dd/MM/yyyy").parse("10/06/2021");
var pedido = new Pedido(fecha, "Cesar Díaz");
pedido.adicionarVehiculo(new Automovil("BMW", "i8", 300_000_000d,
                                       TipoTransmision.MANUAL, true, true));
pedido.adicionarVehiculo(new Motocicleta("Suzuki", "VStrong", 30_000_000d, 600));

System.out.printf("Cliente: %s %n", pedido.getCliente());
System.out.printf("Numero vehiculos: %d %n", pedido.calcularCantidadVehiculos());
System.out.printf("Subtotal: %,.2f %n", pedido.calcularSubtotal());
System.out.printf("Descuento: %,.2f %n", pedido.calcularDescuentos());
System.out.printf("Impuestos: %,.2f %n", pedido.calcularImpuestos());
System.out.printf("Total: %,.2f %n", pedido.calcularTotal());
```

Salida

```
Cliente: Cesar Díaz
Numero vehiculos: 2
Subtotal: 334.400.000,00
Descuento: 1.550.000,00
Impuestos: 33.285.000,00
Total: 366.135.000,00
```

Prueba 2

```
var fecha = new SimpleDateFormat("dd/MM/yyyy").parse("17/03/2021");
var pedido = new Pedido(fecha, "Carlos Perez");
pedido.adicionarVehiculo(new Automovil("Kia", "Rio Sedan", 60_000_000d,
                                       TipoTransmision.MANUAL, false, false));
pedido.adicionarVehiculo(new Automovil("Kia", "Rio Hatchback", 64_000_000d,
                                       TipoTransmision.AUTOMATICA, true, true));
pedido.adicionarVehiculo(new Motocicleta("Suzuki", "VStrong", 30_000_000d, 600));
pedido.adicionarVehiculo(new Motocicleta("Auteco", "VICTORY BOLD", 5_999_000d, 125));

System.out.printf("Cliente: %s %n", pedido.getCliente());
System.out.printf("Numero vehiculos: %d %n", pedido.calcularCantidadVehiculos());
System.out.printf("Subtotal: %,.2f %n", pedido.calcularSubtotal());
System.out.printf("Descuento: %,.2f %n", pedido.calcularDescuentos());
System.out.printf("Impuestos: %,.2f %n", pedido.calcularImpuestos());
System.out.printf("Total: %,.2f %n", pedido.calcularTotal());
```

Salida

```
Cliente: Carlos Perez
Numero vehiculos: 4
Subtotal: 167.599.000,00
Descuento: 0,00
Impuestos: 10.160.000,00
Total: 177.759.000,00
```

Esqueleto:

```
public class Pedido {

    public void adicionarVehiculo(Vehiculo vehiculo) {
        // TODO: Implementar
    }

    public Integer calcularCantidadVehiculos() {
        // TODO: Implementar
        return null;
    }

    public Double calcularSubtotal() {
        // TODO: Implementar
        return null;
    }

    public Double calcularDescuentos() {
        // TODO: Implementar
        return null;
    }

    public Double calcularImpuestos() {
        // TODO: Implementar
        return null;
    }

    public Double calcularTotal() {
        return calcularSubtotal()-calcularDescuentos()+calcularImpuestos();
    }
}

public abstract class Vehiculo {

    public abstract Double calcularPrecio();

    public abstract Double calcularPorcentajeImpuesto();

    public Double calcularDescuento(Date fecha) {
        // Extrae el mes de la fecha actual.
        // Para comparar use las constantes Calendar.JULY o Calendar.JUNE
        var cal = GregorianCalendar.getInstance();
        cal.setTime(fecha);
        var mes = cal.get(Calendar.MONTH);
```

```
        // TODO: Implementar
        return null;
    }
}

public class Automovil {

    public Double calcularPrecio() {
        // TODO: Implementar
        return null;
    }

    public Double calcularPorcentajeImpuesto() {
        // TODO: Implementar
        return null;
    }

}

public class Motocicleta {

    public Double calcularPrecio() {
        // TODO: Implementar
        return null;
    }

    public Double calcularPorcentajeImpuesto() {
        // TODO: Implementar
        return null;
    }

}

public enum TipoTransmision {
    MANUAL, AUTOMATICA
}
```

Recuerda: Implementa las clases y las relaciones de manera correcta. Cada método en el modelo de datos debe ser implementado para el correcto funcionamiento de la aplicación.

Nota: Recuerde que cada clase debe ser codificada en un archivo independiente, pero se deben de cargar juntas en iMaster (no incluir el ***package***).