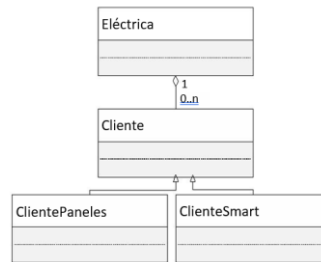




Un proveedor energía eléctrica quiere disponer de una aplicación que le permita gestionar el consumo y realizar la tarificación de sus clientes. Para ello se ha de implementar una aplicación Orientada a Objetos cuyo diseño se ilustra simplificado en la siguiente figura:



- **Cliente.** Modela al cliente clásico de la compañía eléctrica que tiene un contador de consumo de energía eléctrica.
- **ClientePaneles.** Modela a un tipo especial de cliente que tiene instalados paneles solares, el cual, además del contador de consumo de energía eléctrica, dispone de un contador que mide la energía producida por sus paneles solares que es inyectada a la red de la compañía.
- **ClienteSmart.** Modela a otro tipo especial de cliente que tiene un descuento en la tarifa.
- **Eléctrica.** Modela a la compañía eléctrica.

Implementa la clase **Cliente** teniendo en cuenta que un cliente se caracteriza por los siguientes atributos:

- El dni del cliente. Cadena de caracteres.
- El **contador de energía consumida** (kWh). Última lectura del contador. Número real.
- La **energía consumida** (kWh). Número real. Indica la energía consumida en el mes actual.
- La **potencia contratada** (kW). Número real.

Y las siguientes operaciones:

- Un **constructor** que tiene como parámetros el dni y la potencia contratada. Inicializa además la energía consumida y el contador de energía consumida a 0.0.
- Un método **void actualizaEnergíaConsumida(double nuevaLecturaContador)** que calcula la energía consumida en el mes actual restando el valor del parámetro nuevaLecturaContador del valor del contador energía consumida (la última lectura que es la del mes anterior). Además, actualiza el contador de energía consumida.
- Un método **double importeFactura()** que devuelve el importe total en euros a ser facturado en el último periodo al cliente. Se utiliza una facturación por tramos en el que se suma a un término fijo (distinto para cada tramo) el importe de la energía consumida de acuerdo a la fórmula:

Importe= costo del término fijo del tramo + energía consumida*coste del kWh

- Para potencia contratada < 3,4 kW: Costo del término fijo 5€ y Costo del kWh: 0,00138€
- 3,4 kW <= Potencia contratada <= 6,4 kW. Costo del término fijo 7€ y Costo del kWh: 0,00276€
- potencia contratada > 6,4 kW. Costo del término fijo 9€ y Costo del kWh: 0,01104€

Implementa la clase **ClientePaneles** que tiene además los atributos:

- **Energía inyectada:** Número real. Energía eléctrica, medida en kWh, producida mensualmente por los paneles del cliente, que es inyectada a la red de la compañía eléctrica.
- **Contador de energía inyectada.** Número real. Es el último valor leído del contador de energía inyectada medida en kWh.

Y las siguientes operaciones:

- Un **constructor** que tiene como parámetros el dni y la potencia contratada. Inicializa además la energía consumida, el contador de energía consumida, la energía inyectada y el contador de energía inyectada a 0.0.
- Un método **void actualizaEnergiaInyectada(double nuevaLecturaContador)**. que calcula la energía inyectada en el mes actual restando el valor del parámetro nuevaLecturaContador del valor del contador energía inyectada (la última lectura que es la del mes anterior). Además, actualiza el contador de energía inyectada.
- Un método **double importeFactura()** que devuelve el importe mensual a facturar en euros, calculándose de la siguiente manera:

$$\text{importe} = \text{importe de la factura sobre la energía consumida} - (\text{energía inyectada} * 0,0034).$$

Implementa la clase **ClienteSmart** que tiene además el atributo:

- **Porcentaje de descuento.** Número real que indica para cada ClienteSmart el porcentaje (en tanto por ciento) de descuento que se le aplica en la tarificación.

Y las siguientes operaciones:

- Un **constructor** que tiene como parámetros el dni, la potencia contratada y porcentaje de descuento. Inicializa además la energía consumida y el contador de energía consumida a 0.0.
- Un método **double importeFactura()** que devuelve el importe a facturar en el último periodo en euros, calculado sobre importe de la factura de la energía consumida del cliente, aplicando el descuento que se indica en porcentaje de descuento.

Implementa la clase **Eléctrica** que se caracteriza por el atributo:

- **clientes** que es una colección de clientes de la empresa

Y las siguientes operaciones:

- Un constructor que no tiene parámetros. Inicializa su único atributo.
- Un método **void agregaCliente(Ciente nuevoCliente)** que agrega el cliente pasado como parámetro a la colección de clientes de la compañía eléctrica.
- Un método **int getCantidadClientesPaneles()**. Devuelve el número de clientes con paneles que tiene la compañía.
- Un método **double getFacturacionTotal()**. Devuelve el importe total a facturar a todos los clientes de la compañía.

Implementa la clase **PruebaElectrica** con un método **main()** que cree la compañía eléctrica agregando al menos 3 clientes: uno básico, uno de paneles y uno smart, y pruebe los métodos anteriores.