11076 - Programación Orientada a Objetos
Test del primer parcial

SISTEMA DE FACTURACION

Una cooperativa eléctrica pide que se le desarrolle un nuevo sistema **facturador**. El sistema debe permitir dar alta de medidores, ingresar el estado de un medidor, y emitir facturas. Tenga en cuenta los siguientes detalles:

- Por cada medidor se conoce el número de medidor, la dirección donde está instalado (número de calle y número de casa) y los datos del titular (nombre, teléfono y la dirección a la que debe enviarse la factura).
- Por cada medidor se conocen las dos últimas mediciones (todas las mediciones anteriores se olvidan).
- Al facturar, se toma en cuenta la diferencia entre las dos mediciones o sea, Kw de la ultima medición menos Kw de la medición anterior (eso seria el consumo a facturar).
- Hay dos tipos de medidores, los reactivos y los reactivos-inductivos. De los medidores reactivos solo se tienen mediciones de corriente reactiva. De los reactivos-inductivos se tienen mediciones de corriente reactiva y además mediciones de corriente inductiva. (es decir, las dos ultimas de cada tipo de corriente).
- El Kilovatio de corriente reactiva cuesta $1.
- Hasta 1000 Kilovatios de corriente inductiva son gratis; cada Kilovatio extra cuesta $2.

El sistema debe permitir:

- Dar de alta un medidor (con su titular).
- Dado un número de medidor, recuperar el medidor correspondiente.
- Ingresar valores medidos para un medidor determinado (dado que solo se mantienen dos, la mas antigua se pierde).
- Facturar un medidor determinado creando un objeto Factura que entiende los mensajes
- nombreTitular(), numeroMedidor(), direccionTitular() y monto(). Tenga en cuenta que la respuesta al mensaje direccionTitular() debe reflejar la dirección actual del titular aún si esta cambia después de crear el objeto Factura. Además, tenga en cuenta que si no existen mediciones, el monto facturado debe ser 0.
- Facturar todos los medidores (retorna una colección de facturas).

1. Construya un diagrama UML de clases donde quede claro que clases componen el sistema, asociaciones, herencia, y atributos. No es necesario indicar métodos.
2. Implemente las clases necesarias para proveer la funcionalidad antes descripta.
3. Muestre en un caso de prueba de JUnit como inicializar el sistema, como dar de alta un medidor de cada tipo para un mismo titular (uno para su domicilio particular y otro para su empresa), como ingresar mediciones para uno de los medidores, y como facturar todos los medidores.