

36019475. CSIFC03. MP0485. Programación

**XUNTA DE GALICIA**CONSELLERÍA DE CULTURA, EDUCACIÓN
E ORDENACIÓN UNIVERSITARIA

Páxina principal ► Os meus cursos ► Formación Profesional a Distancia ► Curso 2016-2017 ►
36019475 IES de Rodeira ► CSIFC03 Desenvolvemento de aplicacións web ►
125_36019475_ZSIFC03_MP0485_A ► Xeral ► Indicacións e Explicacións á solución do exame.

NAVEGACIÓN

Páxina principal

• Amiña área persoal

Páxinas do sitio

O meu perfil

Curso actual

125_36019475_ZSIFC03_MP0485_A

Participantes

Distincións

Xeral

Foro de noticias.

Foro de presentación.

Foro de cafetería.

Tutoría por chat.

Guía del alumnado.

Horario y Criterios de Evaluación Generales

Recursos generales.

Videotutoriais de Java

Glosario.

Notas primera evaluación.

Enunciado do Exame da Primeira Avaliación

Indicacións e Explicacións á solución do exame.

Solucións

Notas segunda evaluación.

Enunciado e Solución do Exame

Notas da terceira avaliación

Exame da 3ª Avaliación

EXERCICIO 1

EXERCICIO 2 - Solución

EXERCICIO 2 - Solución con Clases Auxiliares

EXERCICIO 3

EXERCICIO 3 - Con "Simulación" da clase DropBoxPre...

EXERCICIO 4

Notas finais.

Exame Final

Solucións Exame Final

Foro de Exame do 2 Trimestre

Enunciado Xeral dos Exames

Ficheiros Auxiliares de Axuda

Exame Tipo 1

Enunciado do Exame 1

Solución

Enunciado do Exame Tipo 2

Solución ao Apartado 1

Solución ao apartado 2

Exercicio de Repaso

Unidade didáctica 1

Unidade didáctica 2

Unidade didáctica 3

Unidade didáctica 4

Unidade didáctica 5

Unidade didáctica 6

Unidade didáctica 7

Unidade didáctica 8

Unidade didáctica 9

Unidade didáctica 10

Unidade didáctica 11

Os meus cursos

ADMINISTRACIÓN

Administración do curso

Configuración do meu perfil

Indicacions e Explicacións á solución do exame.

O exame pedía:

1.- Implementar unha aplicación que visualice un menú coas seguintes opcións:

- **Alta de Cliente**
- **Acceso ós datos de Consumo**

O Alta de cliente debe: Crear un cliente pedindo os seus datos e unha password, crear un Suministro para ese cliente e gardalo na base de datos.

O Acceso ós datos de Consumo debe pedir o número de suministro e a password e visualizar os datos de consumo para ese cliente pedidos no exercicio.

```
import java.util.Scanner;
import com.iesrodeira.consumo.Cliente;
import com.iesrodeira.consumo.Suministro;
import com.iesrodeira.consumo.SuministroException;
import com.iesrodeira.consumo.CCCEException;
import com.iesrodeira.consumo.DNIException;
```

```
class InformacionCliente {

    public static void altaCliente() {

        String password;
        String tarifa;
        float potencia;

        try {
            Cliente cl=Cliente.newClienteConsole();
            System.out.println("Tarifa?: ");
            tarifa=scn.nextLine();
            System.out.println("Potencia Contratada?: ");
            potencia=Float.parseFloat(sc.nextLine());
            System.out.println("Introduce a password: ");
            password=scn.nextLine();
```

```

System.out.println("Cliente dado de alta : "+cl+" (Código de Suministro: "+sum.getCodigo()+")");
} catch (NumberFormatException e) {
    System.out.println("A potencia debe ser un número decimal en KW");
} catch (CCCEException e) {
    System.out.println(e.getMessage());
} catch (DNIException e) {
    System.out.println(e.getMessage());
} catch (SuministroException e) {
    System.out.println(e.getMessage());
}
}

public static void datosConsumo() {
    String nums,password;
    Suministro sum;

    try {
        System.out.println("Introduce o código de Suministro: ");
        nums=scn.nextLine();
        System.out.println("Introduce a password: ");
        password=scn.nextLine();
        sum=Suministro.getInstance(nums,password);
        System.out.println("Datos do Suministro: "+sum.getCliente());
        System.out.println("Nº Suministro: "+sum.getCodigo());
        System.out.println("Potencia Contratada: "+sum.getPotencia());
        System.out.println("Tipo de Tarifa: "+sum.getTarifa());
        System.out.println("Consumo en KW/h: "+sum.getConsumo());
        System.out.println("Importe Factura: "+sum.getImporteFactura());
    } catch (SuministroException e) {
        System.out.println(e.getMessage());
    }
}

public static void main(String[] args) {
    int op;

    Scanner scn=new Scanner(System.in);

    do {
        try {
            System.out.println("1.- Alta de Cliente");
            System.out.println("2.- Acceso ós datos de consumo");
            System.out.println("3.- Saír");
            String line=scn.nextLine();
            op=Integer.parseInt(line);
        } catch (NumberFormatException e) {
            op=0;
        }

        switch(op) {
            case 1: altaCliente(); break;
            case 2: ic.datosConsumo(); break;
        }
    } while(op!=3);
}
}

```

Como se pode ver, se fai uso dos métodos ofrecidos pola clase Suministro, como si xa estiveran implementados e realizaran exactamente a función que di a documentación. Unha clase é como unha caixa negra que expón a súa funcionalidade a través dos seus métodos e atributos públicos. Se sodes capaces de utilizar os métodos da clase Scanner, System, Integer, Float, String ... etc, ¿por qué non os da clase Suministro?

2.- Implementar a clase Suministro

Esta clase é algo máis complexa que o programa principal, pero non demasiado, xa que faremos uso dos métodos (APIS) ofrecidos polas clases auxiliares. En primeiro lugar definimos os atributos da clase:

```
private Cliente cliente;
private String codigoSuministro;
protected String tarifa;
protected float potencia;
float ultimaLectura;
public float lecturaActual;
private String password;
private float preciokw;
```

Constructor: Debe inicializar o obxecto co valor da última lectura e gardalo na BBDD.

```
public Suministro(Cliente cl,String tarifa,float potencia,String passwd) throws SuministroException {
    // Obtemos a data para crear o código de suministro

    Calendar cal=Calendar.getInstance();
    int ano=cal.get(Calendar.YEAR);
    int mes=cal.get(Calendar.MONTH);
    int dia=cal.get(Calendar.DAY_OF_MONTH);

    String strmes;
    String strdia;

    strmes=String.valueOf(mes+1);
    strdia=String.valueOf(dia);
    if (mes<10) strmes="0"+strmes;
    if (dia<10) strdia="0"+dia;

    // Almacenamos a información nos atributos
    try {
        this.cliente=cl; // OLLO!!! Non fai unha copia.....
        this.password=passwd;
        this.potencia=potencia;
        this.codigoSuministro=ano+strmes+strdia+cl.getDNI()+tarifa+getLetraPotencia();
        this.tarifa=tarifa;
        preciokw=getPrecioKw(); // Obtemos o precio por Kw, segundo a potencia contratada e a tarifa
        elexida
        ultimaLectura=getUltimoConsumo(); // Obtemos o consumo do contador
        lecturaActual=ultimaLectura;
        SuministroDB db=new SuministroDB(); // Gardamos o Suministro na BBDD
        db.saveSuministro(this,this.password);
    } catch (DBException e) {
        throw new SuministroException(e.getMessage());
    }
}
```

Como se ve facemos uso de dous métodos getLetraPotencia, getPrecioKw e getUltimoConsumo. Os programamos.

```
/** Devolve o precio por KW según a tarifa
 */
```

```
private float getPrecioKw() throws SuministroException {
    float preciokw=0.0f;

    if (tarifa.equals("TUR")) preciokw=0.007343f;
    else if (tarifa.equals("2TR")) preciokw=0.008123f;
    else if (tarifa.equals("")) preciokw=0.011210f;
    else throw new SuministroException("A tarifa non existe");
    return preciokw;
}
```

```
public float getUltimoConsumo() {
    Contador c=new Contador(getCodigo()); // Contactamos co contador usando o código de suministro
    lecturaActual=c.getLectura(); // Obtemos a lectura do contador
    return lecturaActual;
}
```

```
/** Devolve a letra correspondente á potencia contratada
```

```

private char getLetraPotencia() {
    char letraPotencia='D';

    if (potencia <= 2.3) letraPotencia='A';
    else if (potencia <= 3.5) letraPotencia='B';
    else if (potencia <= 4.6) letraPotencia='C';
    return letraPotencia;
}

```

getInstance: devolve o obxecto Suministro correspondente co código de suministro si a password é correcta

```

public static Suministro getInstance(String codSuministro,String passwd) throws SuministroException {
    Suministro s=null;
    try {
        SuministroDB db=new SuministroDB(); // Conectamos coa BDD e recuperamos o Suministro
        s=db.loadSuministro(codSuministro,passwd);
    } catch (DBException e) {
        throw new SuministroException(e.getMessage());
    }
    return s;
}

```

procesaFactura: actualiza a información de consumo do Suministro na BDD. NON PODE SER PUBLIC

xa que o enunciado dice que únicamente poderán utilizalo os membros do paquete. (erro na solución subida)

```

void procesaFactura() throws SuministroException {
    try {
        SuministroDB db=new SuministroDB(); // Obtemos o último consumo e actualizamos a BBDD
        getUltimoConsumo();
        ultimaLectura=lecturaActual;
        db.saveSuministro(this,this.password);
    } catch (DBException e) {
        throw new SuministroException(e.getMessage());
    }
}

```

getImporteFactura, devolve o importe da factura si se facturara hoxe.

```

float getImporteFactura() {
    // O importe se calcula sumando ao factor de potencia o importe do consumo
    return getFactorPotencia()+getConsumo()*preciokw;
}

```

Como vemos, utilizamos un método auxiliar getFactorPotencia(), o implementamos:

```

/**
Devolve o factor de potencia a aplicar á factura según a potencia contratada
*/
private float getFactorPotencia() {
    float factor=0.0f;
    char letra=getLetraPotencia();
    switch(letra) {
        case 'A': factor=3.17f; break;
        case 'B': factor=4.23f; break;
        case 'C': factor=5.12f; break;
        case 'D': factor=7.06f; break;
    }
    return factor;
}

```

getUltimoConsumo - Obten do contador o valor do consumo do contador (consumo total).

```

public float getUltimoConsumo() {

```

```
}  
  
getConsumo(); - Obten o consumo dende a última factura á actualidade en kw  
  
public float getConsumo() {  
    getUltimoConsumo(); // O Consumo é a lectura actual do contador - a lectura anterior  
    return (lecturaActual-ultimaLectura);  
}
```

Última modificación: Mércores, 14 de Decembro do 2016, 17:50