

# ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

PROGRAMACION ORIENTADA A OBJETOS

TERCERA EVALUACIÓN – I TERMINO 2012-2013

Nombre: \_\_\_\_\_ Matrícula \_\_\_\_\_ Paralelo \_\_\_\_\_

## **TEMA 1. (5 puntos)**

Conteste las siguientes preguntas con VERDADERO (V) o FALSO (F) según correspondan:

- a) Una clase abstracta solo puede implementar una interfaz ( )
- b) El bloque finally se ejecuta siempre que se capture una excepción ( )
- c) El operador new crea nuevas instancias en una enumeración ( )
- d) Al cambiar el valor de una variable de instancia, este cambia solo en la instancia actual y no en el resto de instancias ( )
- e) Una clase anónima solo es instanciada una sola vez ( )

## **TEMA 2. (10 puntos)**

Que imprime el siguiente código:

<pre>class X {     String getA(String s){         return s;     } } class A extends X {     String getA(){ return "A";}     String getA(String s){ return "C";} }</pre>	<pre>class B extends A {     String getA(){ return "B";}     void test(){         A a = this;         System.out.print(this.getA());         System.out.print(a.getA());         System.out.print(((A)this).getA());         System.out.print(super.getA());         System.out.print(this.getA("B"));     }     public static void main(String[] args){         new B().test();     } }</pre>
---	--

Respuesta:

## **TEMA 3. (10 puntos)**

Que imprime el siguiente código:

<pre>public class Test{     public static void main(String[] args){         B b = new B();         System.out.println(b.i);         System.out.println(b.j);         A a = new B();         System.out.println(a.i);         System.out.println(a.j);         a.m();         a.m1();         System.out.println(a.i);         System.out.println(a.j);     } }</pre>	<pre>class A{     int i=1;     static int j=3;     void m(){ i=5; }     static void m1(){ j=7;} } class B extends A{     int i=2;     static int j = 4;     void m(){ i=6; }     static void m1(){ j=8; } }</pre>
--	---

Respuesta:

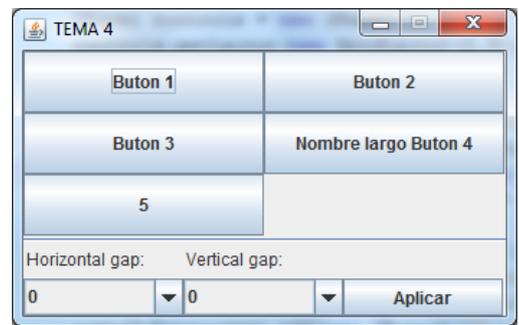
### TEMA 4. (15 puntos)

Para crear la siguiente interfaz agregue los controles y layouts en el orden correcto

```
import java.awt.*;
import javax.swing.*;
public class Ventana extends JFrame {
.....
public void addComponentsToPane(final Container pane) {
    JComboBox horGapComboBox = new JComboBox(new String[]{"0", "10", "15", "20"});
    JComboBox verGapComboBox = new JComboBox(new String[]{"0", "10", "15", "20"});
    final JPanel panelBotones = new JPanel();
    panelBotones.setLayout(
    );
    JPanel controles = new JPanel();
    controles.setLayout(
    );
    //Set up components preferred size
    Dimension buttonSize = new JButton("Just fake button").getPreferredSize();
    panelBotones.setPreferredSize(new Dimension((int)(buttonSize.getWidth() * 2.5)+20,
    (int)(buttonSize.getHeight() * 3.5)+40));
    //Agregar botones

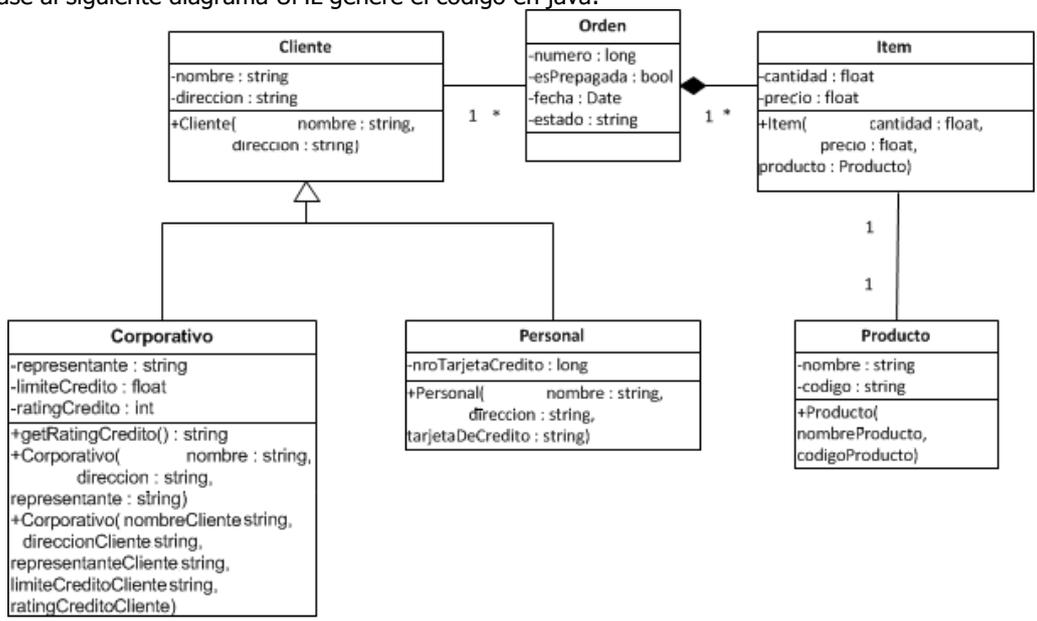
//Agregar controles

//Agregar a la ventana
}
}
```



### TEMA 5. (20 puntos)

En base al siguiente diagrama UML genere el código en java:



## TEMA 6. (40 puntos)

Se solicita implementar un Address Book o Libreta de Direcciones/Contactos el mismo que debe constar de un índice alfabético para almacenar los nombre de los contactos. La información a almacenar será: nombre, apellido, dirección, teléfonos (que pueden ser de tipo convencional y celular, para lo cual se requiere tener un objeto Telefono). Los contactos no se deben repetir en ningún momento y esto debe ser validado por la aplicación, en este caso se debe presentar un mensaje al usuario indicado que el contacto esta repetido e indicar que solamente puede ingresar los números de teléfono. Un contacto puede tener solamente un número de teléfono de un tipo definido.

- Se requiere crear el GUI para el ingreso de los datos y que sean almacenados en algún tipo de estructura al presionar el botón Grabar. Recordar que la funcionalidad de esta libreta de direcciones es similar a un directorio/guía telefónica.
- Implementar el método **buscar (Collection c, String apellido, String nombre)**, el cual busque por el índice el apellido y nombre de la persona y devuelva la información completa del contacto.

```
import javax.swing.*;
import java.awt.*;
public class Pantalla extends JFrame {
    public Pantalla() {
        super("Contactos");

        .....
        setSize(300, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
public class Directorio {
    public static void main(String[] args) {
        Pantalla mf = new Pantalla();
        mf.show();
    }
}
```

```
JRadioButton
JRadioButton();
JRadioButton(Icon icon)
JRadioButton(String
text);
JRadioButton(String
text,boolean selected);
Boolean is Selected();
Set keySet();
Collection values()
```