

Primer Certamen

Todas las preguntas tienen igual puntaje.

1. a) Un diseñador propone crear la clase Punto como clase heredada de Circulo de radio cero. Dé dos razones por la que éste sería un mal diseño.
- b) Indique dos semejanzas y dos diferencias **entre** crear una clase abstracta y crear una interfaz.
- c) ¿Qué es ligado dinámico? Dé un ejemplo.

a) i) Es un mal diseño porque no cumple el principio de sustitución de buena forma. Si cuando esperamos un círculo y por sustitución pasamos un punto (dado que en este diseño un punto es un círculo), podríamos cambiar el radio del círculo; sin embargo por tratarse un punto probablemente esto no ocurriría y no entenderíamos por qué nuestro cambio de radio no se efectúa.

ii) Es un mal diseño porque no hace buen uso de memoria. Para almacenar un punto sólo se requiere definir dos valores en R2, si heredamos de círculo estaremos usando más memoria que la necesaria.

b) Semejanzas: i) En ambos casos aplica el principio de sustitución tanto para quienes hereden de la clase o implementen la interfaz. ii) En ambos casos tenemos métodos que deben ser implementados.

Diferencias: i) Se puede heredar sólo de una clase abstracta, pero se puede implementar múltiples interfaces. Es decir si queremos usar el principio de sustitución para la clase abstracta en una clase que ya hereda de otra clase, estamos mal. Esto no crea problemas en el caso de interfaces. ii) La interfaz no puede incorporar atributos no estáticos y constantes, la clase abstracta sí.

- c) **Ligado dinámico es la definición en tiempo de ejecución del código asociado a un cierto llamado o invocación. Se liga el nombre de la invocación con el método específico a llamar en tiempo de ejecución y no de compilación.**

Ejemplo: Como el visto en clases o alguno similar.

```
abstract class Animal
{
    abstract void print();
}

class Cat extends Animal {
    private int catNumber;
    Cat(int i) {
        catNumber = i;
    }
    void print() {
        System.out.println("Cat #" + catNumber);
    }
}

class Dog extends Animal{
    private int dogNumber;
```

```

Dog(int i) {
    dogNumber = i;
}
void print() {
    System.out.println("Dog #" + dogNumber);
}
}

public class CatsAndDogsV2 {
    public static void main(String[] args) {
        ArrayList animals = new ArrayList();
        for(int i = 0; i < 7; i++)
            animals.add(new Cat(i));
        // Not a problem to add a dog to cats:
        animals.add(new Dog(7));
        for(int i = 0; i < animals.size(); i++)
            ((Animal)animals.get(i)).print();
    }
}

```

2. a) ¿Si el archivo que contiene el siguiente programa se llama p1.java, con qué comando de consola lo compilaría?
- b) ¿Con qué comando lo ejecutaría?
- c) ¿Qué se muestra por pantalla tan pronto se ejecuta?
- d) ¿Qué hace el programa?
- e) ¿Re-escriba la parte que corresponda para eliminar toda clase interna o anónima.

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;

class Mimic extends JFrame {
    MimicGUI gui;
    public Mimic() {
        gui=new MimicGUI(this);
        setTitle("Mimic");
        setSize( 250, 100);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        getContentPane().add(gui);
        setVisible(true);
    }
    public static void main( String[] args) {
        Mimic mimic = new Mimic();
    }
}

class MimicGUI extends JPanel {
    private JLabel label = new JLabel(" Echo appears here");
    private JTextField quote = new JTextField( 20);
    private JFrame frame;
    public MimicGUI(JFrame f) {
        frame=f;
        add(quote);
        add(label);
        quote.addActionListener(
            new ActionListener() {

```

```

        public void actionPerformed( ActionEvent e) {
            frame.setTitle(label.getText());
            label.setText(quote.getText());
        }
    }
};
}
}

```

a) % javac p1.java

b) % java mimic

c)



d) Cada vez que el usuario escribe en el campo de texto y presiona enter, el texto del label o rótulo es copiado como nombre del frame y el nuevo texto ingresado en el campo de texto es copiado en el label. El valor del campo de texto no es borrado. Al presionar la x del frame se cierra la aplicación.

e) La versión sin clases anónimas ni internas de este programa es:

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;

class Mimic extends JFrame { // aqui no hay clases internas
    MimicGUI gui;
    public Mimic() {
        gui=new MimicGUI(this);
        setTitle("Mimic");
        setSize( 250, 100);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        getContentPane().add(gui);
        setVisible(true);
    }
    public static void main( String[] args) {
        Mimic mimic = new Mimic();
    }
}

class MimicGUI extends JPanel {
    private JLabel label = new JLabel(" Echo appears here");
    private JTextField quote = new JTextField( 20);
    private JFrame frame;
    public MimicGUI(JFrame f) {
        frame=f;
        add(quote);
        add(label);
        quote.addActionListener(new TextoActionListener(frame,label,quote));
    }
}

// Aquí está la diferencia. Una clase anónima se hace explícita.
class TextoActionListener implements ActionListener {

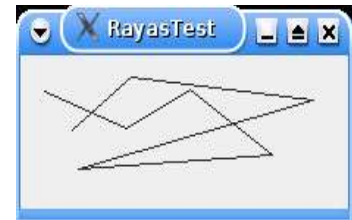
```

```

private JFrame myFrame;
private JLabel myLabel;
private JTextField myQuote;
public TextoActionListener(JFrame frame, JLabel label, JTextField quote) {
    myFrame = frame;
    myLabel = label;
    myQuote = quote;
}
public void actionPerformed( ActionEvent e) {
    myFrame.setTitle(myLabel.getText());
    myLabel.setText(myQuote.getText());
}
}

```

3. Escriba una aplicación Java que permita dibujar rayas rectas con el mouse. Cada vez que se presiona el mouse se agrega a lo ya hecho una línea entre el último punto y el actual.



```

import java.awt.*;
import java.awt.event.*;
import java.util.*;
import java.awt.geom.*;
import javax.swing.*;

public class RayasTest
{
    public static void main(String[] args)
    {
        RayasFrame frame = new RayasFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.show();
    }
}

class RayasFrame extends JFrame
{
    public RayasFrame()
    {
        setTitle("RayasTest");
        setSize(250, 150);

        RayasPanel panel = new RayasPanel();
        Container contentPane = getContentPane();
        contentPane.add(panel);
    }
}

class RayasPanel extends JPanel
{
    public RayasPanel()
    {
        puntos = new ArrayList();
        addMouseListener(new MouseHandler());
    }

    public void paintComponent(Graphics g)

```

```
{
    super.paintComponent(g);
    Graphics2D g2 = (Graphics2D)g;

    for (int i = 0; i < puntos.size()-1; i++)
        g2.draw(new Line2D.Double((Point)puntos.get(i),
                                   (Point)puntos.get(i+1)));
}

private ArrayList puntos;

private class MouseHandler extends MouseAdapter
{
    public void mouseClicked(MouseEvent event)
    {
        puntos.add(event.getPoint());
        repaint();
    }
}
}
```