

Primer Certamen: Inicio: 12:00 hrs, término: 13:40 hrs.

Todas las preguntas tienen igual puntaje. No se atienden preguntas. Si estima que algo no se entiende o está ambiguo, haga una suposición razonable y continúe.

1.- Responda brevemente:

a) Mencione alguna característica de Java que ayuda a la reutilización de software, explique por qué.
Java permite manejar Herencia. Esta característica ayuda a la reutilización de código porque podemos apoyarnos en una clase existente y extenderla para incorporar algún aspecto específico propio de nuestro problema. Ejemplo de ello son las clases de elementos gráficos, como JFrame, botones entre otras, las cuales reutilizamos extendiéndolas según nuestro propósito.

La existencia de Interfaces es otra característica. Ellas permiten crear código genérico dependiente de un conjunto de operaciones bien definidas.

También podemos decir que la encapsulación ofrecida en las clases de los lenguajes orientados a objeto y en particular Java ayuda a la reutilización de software.

b) ¿Qué es una instancia? ¿Qué es una clase? ¿qué es un objeto?

Una instancia es un elemento específico de una categoría de elementos. En el contexto de orientación a objetos, una instancia de una clase y objeto son lo mismo.

Una clase es una abstracción para una categoría de elementos que mantienen características comunes. En orientación a objetos, una clase define un conjunto de atributos y un conjunto de operaciones.

Un objeto es una instancia de una clase. Un objeto posee nombre, un estado almacenado en sus atributos y servicios reflejados en sus métodos.

c) Una método tiene un calificador de acceso, un valor retornado, un nombre y un conjunto de parámetros. ¿Cuáles de éstos identifican en forma unívoca a un método, de manera que el compilador no diga que está definido dos veces?

El nombre y el conjunto de parámetros definen unívocamente un método.

d) Cuando usted corre la aplicación Prueba.class, ¿qué nombre de proceso aparece en la lista de procesos de la máquina (ya sea en windows o Linux)? ¿Por qué?

Aparece java como proceso. Es la máquina virtual la que corre sobre la máquina nativa y el programa Prueba.class sólo existe para la máquina virtual.

e) ¿Por qué desde un método estático, como el método main, no es posible invocar directamente métodos no estáticos de la clase que lo contiene?

Por que los métodos estáticos pueden ser invocados directamente sin la creación de un objeto previo; luego si el objeto no existe, no hay estado y por ende deja sin sentido la invocación de métodos no estáticos que normalmente accederán a atributos inexistente del objeto.

2. Se tiene:

<pre>public class A { private int a; private Point p; /* otros métodos */ }</pre>	<pre>public class B extends A { private float b; /* otros métodos */ }</pre>
---	--

- a) Redefina el método boolean equals(Object obj) de la clase Object en la clase A y luego en clase B.
 b) El constructor copia es aquel que crea un objeto con igual estado a otro de la misma clase pasado como argumento. Por ejemplo, existe el constructor Point(Point po); el cual construye e inicializa un punto en la misma localización especificada por el punto po. Cree el constructor copia para las clases A y B.

Para clase A:

```
import java.awt.*;

public class A {
    private int a;
    private Point p;
    public A() { // esto no se pedia
        a=1;
        p= new Point();
    }
    // constructor copia
    public A (A oa){ // 5 pts
        a=oa.a;
        p=new Point(oa.p);
    }
    // metodo equals
    public boolean equals(Object obj){ // 8 pts.
        if (this == obj) return true;
        if (obj == null) return false;
        if (getClass() != obj.getClass())
            return false;
        A other = (A)obj;
        if ((p==null) return ((other.p==null)&&(a==other.a)); // esta verificacion
falso en clases
        return p.equals(other.p) && a==other.a; // no es penalizada
    }
    // esto no se pedia
    public static void main(String argv[]) {
        A a1 = new A();
        A a2=null;
        if (a1.equals(a2)) System.out.println("a1 y a2 son iguales");
        A a3=a1;
        if (a1.equals(a3)) System.out.println("a1 y a3 son iguales");
        A a4 = new A(a1);
        if (a1.equals(a4)) System.out.println("a1 y a4 son iguales");
        B b= new B();
        if (a1.equals(b)) System.out.println("a1 y b son iguales");
        A a5 = new A(b);
        if (a1.equals(a5)) System.out.println("a1 y a5 son iguales");
    }
    /* otros metodos */
}
```

Para clase B:

```
public class B extends A {
    private float b;

    public B() { // esto no se pedia
        b=1;
    }
    // constructor copia
    public B (B ob){ // 5 pts
        super(ob);
    }
}
```

```

    b=obj.b;
}
// metodo equals
public boolean equals(Object obj){ // 7 pts
    if (!super.equals(obj)) return false;
    B other = (B)obj;
    return b==other.b;
}
// esto no se pedia
public static void main(String argv[]) {
    B b1 = new B();
    B b2=null;
    if (b1.equals(b2)) System.out.println("b1 y b2 son iguales");
    B b3=b1;
    if (b1.equals(b3)) System.out.println("b1 y b3 son iguales");
    B b4 = new B(b1);
    if (b1.equals(b4)) System.out.println("b1 y b4 son iguales");
    A a= new A();
    if (b1.equals(a)) System.out.println("b1 y a son iguales");
}
/* otros metodos */
}

```

3. Un registro de desplazamiento de 8 bits, permite las operaciones:

load: para cargar paralela de cada uno de los 8 bits a través de una variable tipo byte.

reset: para llevar cada bit a cero

shiftLeft: para desplazar los bits a la izquierda, retornando el bit de más a la izquierda e ingresando un bit dato por la derecha

getBits: retorna un byte con los 8 bits.

Cree la clase RegistroDesplazamiento según la especificación dada. Como diseñador e implementador de la clase, decida usted apropiadamente lo que necesite o falte.

```

public class RegistroDesplazamiento {
    private byte bits; // 5 pts.
    public void load (byte dato) { // 5 pts por cada método.
        bits = dato;
    }
    public void reset() {
        bits = 0;
    }
    // las operaciones con bytes cambian los byte a int antes
    // por eso los casts. En la prueba no se penalizan.
    // Me di cuenta aqui.
    public byte shiftLeft(byte in) {
        byte out = (byte) ((bits & 0xFF) >>> 7); // bit de mas a la izquierda
        bits = (byte) (bits << 1);
        bits |= in;
        return out;
    }
    public byte getBits() {
        return bits;
    }
}
// esto no se pedia
public static void main (String argv[]){
    RegistroDesplazamiento rd = new RegistroDesplazamiento();
    rd.load((byte)0x0F);
    System.out.println(rd.getBits());
    System.out.println(rd.shiftLeft((byte)1) + " "+rd.getBits());
}

```

```

    System.out.println(rd.shiftLeft((byte)1) + " "+rd.getBits());
    System.out.println(rd.shiftLeft((byte)1) + " "+rd.getBits());
    System.out.println(rd.shiftLeft((byte)1) + " "+rd.getBits());
    System.out.println(rd.shiftLeft((byte)1) + " "+rd.getBits());
    rd.reset();
    System.out.println(rd.getBits());
}
}

```

4.- Considere el programa adjunto.

- Haga un programa equivalente, pero sin funciones internas ni anónimas.
- A partir del programa dado, cree su versión applet. El applet debe dar la opción a definir el texto del botón a través de la página html.
- Muestre un archivo html simple que incluya su applet con nombre de botón "nuevo boton".

Nota: Usted puede usar secciones del programa original poniendo sólo los números en lugar de copiar las líneas.

```

1 import java.awt.*;
2 import java.awt.event.*;
3 import java.util.*;
4 import javax.swing.*;
5 public class AplicacionCreaBotones extends JFrame {
6     public AplicacionCreaBotones() {
7         setTitle("Crea Botones");
8         setSize(250, 150);
9         CreaBotonesPanel panel = new CreaBotonesPanel();
10        Container contentPane = getContentPane();
11        contentPane.add(panel);
12        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
13        setVisible(true);
14    }
15    public static void main(String[] args){
16        new AplicacionCreaBotones();
17    }
18 }
19 class CreaBotonesPanel extends JPanel {
20     public CreaBotonesPanel(){
21         creaBoton();
22     }
23     private void creaBoton() {
24         JButton boton = new JButton("Crea Boton");
25         boton.addActionListener( new ActionListener (){
26             public void actionPerformed(ActionEvent event) {
27                 creaBoton();
28             };
29         });
30         add(boton);
31         validate();
32     }
33 }

```

a) // 10 pts.

Idem desde 1 a 18, luego:

```

class CreaBotonesPanel extends JPanel {
    public CreaBotonesPanel () {
        creaBoton();
    }
}

```

```

    }
    public void creaBoton() { // cambio
        JButton boton = new JButton("Crea Boton");
        boton.addActionListener( new BotonListener(this)); // cambio
        add(boton);
        validate();
    }
}
class BotonListener implements ActionListener { // nuevo
    private CreaBotonesPanel panel;
    public BotonListener(CreaBotonesPanel panel){
        this.panel=panel;
    }
    public void actionPerformed(ActionEvent event) {
        panel.creaBoton();
    }
}

```

b) versión applet: // 10 pts.

```

import java.awt.event.*;
import java.util.*;
import javax.swing.*;
public class AppletCreaBotones extends JApplet {
    public void init() { // cambio
        CreaBotonesPanel panel = new CreaBotonesPanel(getParameter("texto")); //cambio
        Container contentPane = getContentPane();
        contentPane.add(panel);
    }
}
class CreaBotonesPanel extends JPanel { // cambio parcial
    private String texto; // nuevo
    public CreaBotonesPanel(String texto){ // cambio
        this.texto = texto; // nuevo
        creaBoton();
    }
    private void creaBoton() {
        JButton boton = new JButton(texto);
        boton.addActionListener( new ActionListener (){
            public void actionPerformed(ActionEvent event) {
                creaBoton();
            }
        });
        add(boton);
        validate();
    }
}

```

c) Página html: // 5 pts.

```

<applet code="AppletCreaBotones.class" width=400 height=300>
<param name="texto" value="nuevo boton"/>
</applet>

```