UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID
Facultad de Informática
Escuela Universitaria de Informática
Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software

Cuadernillo de examen

ASIGNATURA	**Laboratorio de Sistemas Operativos Abiertos (Java)**	CÓDIGO	**321**
CONVOCATORIA	**Extraordinaria de Septiembre de 2003**	PLAN DE ESTUDIOS	**1996**
ESPECIALIDAD	**Sistemas**	CURSO	**2002/2003**
TURNO	**Mañana**	CURSO ACADÉMICO	**3º**
CARÁCTER	**Anual**	PROGRAMA	**Ingeniería Técnica**
DURACIÓN APROXIMADA	**2,5 horas**		

APELLIDOS: ..... NOMBRE: .....

GRUPO: ..... NÚMERO DE EXPEDIENTE: .....

Preguntas teórico-prácticas

1. Tipos de datos en Java. Diferencias entre los tipos primitivos y los tipos de referencia. Visibilidad de variables.

2. Diferencias entre las clases String y StringBuffer. Ponga un ejemplo de declaración, consturcción y uso de cada una de ellas. Conversiones de datos primitivos a cadena y viceversa.

3. Significado de las palabras reservadas this y super. Ponga al menos un ejemplo de utilización de cada una de ellas

4. Excepciones en Java. El bloque try. Explique la utilidad de la claúsula throws en un método.

5. Diferencias entre un applet y una aplicación Java con interface gráfica. ¿qué modificaciones habría que hacer en una aplicación con interfaz gráfica para convertirla en un applet?

Documentación aportada en clase por el profesor

Puntuación: 1 punto cada pregunta

Preguntas prácticas

1. Un negocio se dedica a la preparación de ordenadores. Cada ordenador que recoge contiene información sobre el código de la reparación, el código del cliente, la descripción de la avería y el precio de la reparación. Además contiene un campo entero que recoge el estado de la reparación con la siguiente codificación:

- Cuando llega un ordenador al taller se queda pendiente de aprobar el presupuesto y el estado es 0.
- Cuando el cliente acepta el presupuesto el campo estado se pone a 1.
- Cuando el ordenador está reparado el campo estado se pone a 2.

Algunos ordenadores están en garantía, en cuyo caso se añadirá un campo con la fecha de la compra (AAAAMMDD).

a) Codifique la clase ordenador. Tendrá un constructor que permita inicializar el código de la reparación, el código del cliente y la descripción de la avería. También tendrá un método que permita convertir a cadena una representación del ordenador.

b) Codifique una clase derivada para los ordenadores en garantía. En esta clase, el método que convierte el ordenador en cadena añadirá también la fecha de la compra.

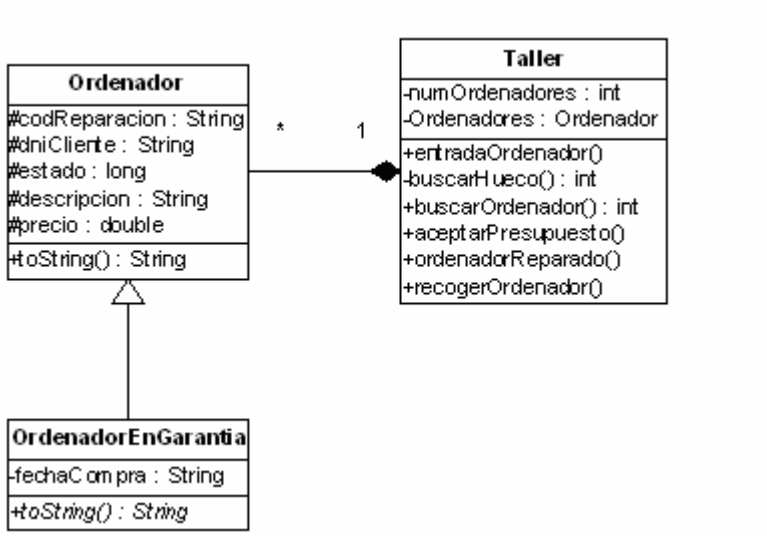
c) Codifique una clase Taller. El constructor de dicha clase permitirá la creación de un array de donde se almacenarán los ordenadores que están en el taller; dicho constructor tomará como argumento el número de ordenadores máximo que puede tener el taller. La clase Taller contendrá métodos para:

- La entrada de un ordenador en el taller.
- La aceptación del presupuesto de una reparación por el cliente. El método deberá buscar el ordenador y poner el estado de la reparación a 1.
- La comunicación del final de la reparación del ordenador. Este método recibirá el código de la reparación y pondrá el estado a 2. En el caso de ordenadores que no estén en garantía recibirá también el precio y actualizará el campo correspondiente.

- La entrega de un ordenador, dejando el elemento que ocupaba a nulo.

Nota importante: La codificación de las clases y métodos se deberá realizar teniendo en cuenta la segunda pregunta práctica de forma que se puedan aprovechar los métodos ya realizados.

Puntuación: 3 puntos.




```
class Ordenador{
    protected String codReparacion = "000000";
    protected String dniCliente = "000000000";
    protected int estado = 0;
    protected String descripcion;
    protected double precio;

    Ordenador(String cod,String cli,String desc){
        codReparacion = cod;
        dniCliente = cli;
        descripcion = desc;
    }

    public String toString(){
        return codReparacion + " " + dniCliente + " " +
               descripcion + " " + estado + " " + precio;
    }

}

class OrdenadorEnGarantia extends Ordenador{
    private String fechaCompra = "aaaammdd";

    OrdenadorEnGarantia(String cod, String cli,String desc,String fecha){
        super(cod,cli,desc);
        fechaCompra = fecha;
    }

    public String toString(){
        return super.toString() + " " + fechaCompra;
    }

}

class Taller{
    int numOrdenadores = 1;
    Ordenador ordenadores[];
```

```
    Taller(int n){
        numOrdenadores = n;
        ordenadores = new Ordenador[n];
    }

    public void entradaOrdenador(Ordenador o){
        int i = buscarHueco();
        ordenadores[i] = o;
    }

    private int buscarHueco(){
        int i = 0;
        while(ordenadores[i] != null && i < numOrdenadores)
            i++;
        return i;
    }

    public int buscarOrdenador(String id){
        for(int i=0;i<numOrdenadores;i++){
            if(ordenadores[i] != null)
                if(ordenadores[i].codReparacion.compareTo(id)==0)
                    return i;
        }
        return -1;
    }

    public void aceptarPresupuesto(String id){
        int i = buscarOrdenador(id);
        ordenadores[i].estado = 1;
    }

    public void ordenadorReparado(String id,double pre){
        int i = buscarOrdenador(id);
        ordenadores[i].precio = pre;
        ordenadores[i].estado = 2;
    }

    public void ordenadorReparado(String id){
        int i = buscarOrdenador(id);
        ordenadores[i].precio = 0;
        ordenadores[i].estado = 2;
    }

    public void recogerOrdenador(String id){
        int i = buscarOrdenador(id);
        ordenadores[i] = null;
    }
}
```

2. Utilizando las clases y métodos de la pregunta anterior se desea realizar una aplicación gráfica que gestione algunas de las tareas de los ordenadores que no están en garantía. La interfaz gráfica tendrá dos cuadros de texto donde el usuario introducirá el código de la reparación y dos botones (la distribución de los controles en la interfaz se dejará a elección del alumno, pudiendo hacerla de la forma más simple que desee).

Las acciones a realizar serán las siguientes:

- Al pulsar sobre el botón de Aceptar presupuesto se invocará el método correspondiente de la clase Taller, buscando el código de reparación que aparezca en el cuadro de texto (no se considerára la posibilidad de que la reparación no exista).
- Al pulsar sobre el botón Reparado se invocará al método correspondiente con el código de reparación y el precio introducido (tampoco se considerará la circunstancia de ordenador no encontrado).

Nota: se supone que al cargar la clase Taller se cargarán todos los ordenadores que actualmente estén en el taller. No es necesario que el alumno realice esta operación.

Puntuación: 2 puntos.

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

public class GestionTaller{
    static Taller t = new Taller(10);
    public static void main(String args[]){
        FrameTaller frm = new FrameTaller();
        //Código para llenar el array de ordenadores
        …
        …
        t.entradaOrdenador(o2);
        t.listarOrdenadores();
        frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frm.show();
    }
}

class FrameTaller extends JFrame{
    JTextField txtCodReparacion = new JTextField();
    JTextField txtPrecio = new JTextField();
    JButton btnAceptar = new JButton("Aceptar presupuesto");
    JButton btnReparado = new JButton("Reparado");
    FrameTaller(){
        setTitle("Gestión del taller");

        getContentPane().setLayout(new GridLayout(4,2));
        getContentPane().add(new JLabel(" Cod. reparacion:"));
        getContentPane().add(txtCodReparacion);
        getContentPane().add(new JLabel(" Precio:"));
        getContentPane().add(txtPrecio);
        getContentPane().add(btnAceptar);
        getContentPane().add(btnReparado);
        pack();

        btnAceptar.addActionListener(new TallerActionListener());
        btnReparado.addActionListener(new TallerActionListener());
    }

    class TallerActionListener implements ActionListener{
         public void actionPerformed(ActionEvent e){
```

```
        //Obtiene el objeto que ha producido el evento
            Object source = e.getSource();
            String cod = txtCodReparacion.getText();
            double pre = Double.parseDouble(txtPrecio.getText());
            if(source == btnAceptar)
                GestionTaller.t.aceptarPresupuesto(cod);
            else
                GestionTaller.t.ordenadorReparado(cod,pre);
            GestionTaller.t.listarOrdenadores();
         }
    }
}
```