

ESCUELA SUPERIOR POLITECNICA DEL LITORAL
FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACION
EXAMEN FINAL - PROGRAMACION ORIENTADA A OBJETOS
I Término 2011, Agosto 2011

Nombre: _____
Matricula: _____

Paralelo: _____

TEMA 1. (20 puntos)

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>class Ventana01 extends JFrame { public Ventana01() { setTitle("Tema 1 - Examen Final POO XD"); setSize(300, 300); // size in pixels setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE) ; } }</pre> | <pre>class Tema1 { public static void main(String[] args) { Ventana01 v = new Ventana01(); Container content = v.getContentPane(); content.add(new Panel1()); v.setVisible(true); } }</pre> |
| <pre>class Panel1 extends JPanel { Color current = Color.red; JButton button = new JButton("Cambiar color"); class ButtonHandler implements ActionListener { public void actionPerformed(ActionEvent e) { current = JColorChooser.showDialog(button, "", Color.white); repaint(); } } class MouseHandler extends MouseAdapter { public void mouseClicked(MouseEvent ev) { current = Color.white; repaint(); } } Panel1() { button.addActionListener(new ButtonHandler()); addMouseListener(new MouseHandler()); add(button); } public void paintComponent(Graphics g) { super.paintComponent(g); Graphics2D g2 = (Graphics2D) g; g.setColor(current); g2.fill(new Rectangle2D.Float(0, 100, 100, 100)); } }</pre> | |

Para cada una de las siguientes preguntas, seleccione la respuesta correcta:

- 1.1 (2 pts.) Qué son ButtonHandler y MouseHandler?**
a) Clases abstractas
b) Clases internas
c) Clases anónimas
d) Clases publica
- 1.2 (2 pts.) JButton y JColorChooser son?**
a) Manejadores de eventos
b) Streams
c) Eventos
d) Controles Gráficos
- 1.3 (2 pts.) La línea super.paintComponent() que hace?**
a) Llama al método paintComponent() de JPanel
b) Llama al método paintComponent() de Ventana01
c) Llama al método paintComponent() de contentpane
d) Dibuja un rectángulo sobre la ventana
- 1.4 (2 pts.) Cual es el color del rectángulo luego de pasar el mouse sobre el panel?**
a) Establece como color por defecto white
b) Establece como color por blue
c) No cambia el color
d) Muestra un cuadro de dialogo para seleccionar el color
- 1.5 (2 pts.)Cuál de los siguiente clases debe ser implementada como clase interna, es decir que si se la coloca fuera de la clase Panel1 no puede trabajar?**
a) ButtonHandler
b) MouseHandler
c) Alternativas a y b
d) Ninguna de las anteriores
- 1.6 (2 pts.) Que es Color.red?**
a) Una variable estatica publica
b) Una variable estatic publica de instancia
c) Un metodo publico estatico
d) Un metodo public de instancia
- 1.7 (2 pts.) Que es paintComponent (en Panel1)?**
a) Una clase interna
b) Un metodo estatico public
c) Un event listener
d) Un metodo de instancia publico
- 1.8 (2 pts.) Que hace repaint en el metodo mouseClicked?**
a) Llama al método paintComponent() de JPanel
b) Llama al método paintComponent() de Ventana01
c) Llama al método paintComponent() de contentpane
d) Repinta el título de la ventana
- 1.9 (2 pts.) Cual es el layout que tiene Ventana01?**
a) BorderLayout
b) FlowLayout
c) NullLayout
d) GridLayout
- 1.10 (2 pts.) Cual es el layout que tiene Panel1?**
a) BorderLayout
b) FlowLayout
c) NullLayout
d) GridLayout

TEMA 2. (30 puntos)

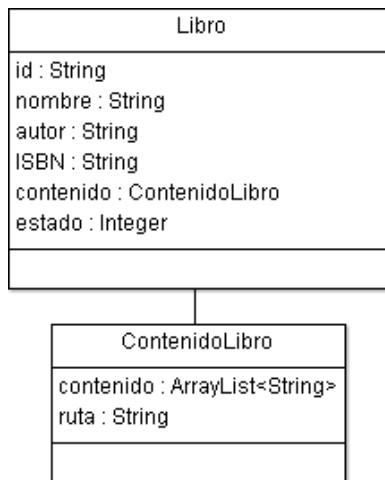
Considere que posee una base de datos basada en archivos, uno de esos archivos contiene la información de los libros que administra una biblioteca. El formato del archivo es el siguiente:

Id@Nombre@Autor@ISBN@Ruta@Archivo@Contenido@Estado@

Dónde:

- Id: Código de 10 dígitos que representa el código del libro para manejo dentro de la biblioteca.
- Nombre: Nombre del libro registrado.
- Autor: Autor principal del libro.
- ISBN: Código universal de 13 dígitos que identifica de manera única un libro.
- Ruta archivo Contenido: Ubicación del archivo en formato txt que tiene un resumen del contenido del libro.
- Estado: El estado del libro: 0 No prestado y 1 prestado.

Considere que tiene las siguientes clases que almacenan la información de un libro:



Implemente los siguientes métodos estáticos en la clase Libro:

- cargarLibros** que recibe la ruta del archivo que contiene los libros de la biblioteca y devuelve los almacena en la colección TreeSet de libros.

Considere que la clase Libro además debe leer la información del contenido de libro almacenada en el archivo deberá leer la información del contenido del libro y almacenarla en el objeto contenido del tipo ContenidoLibro.

- buscarLibros** que realiza una búsqueda libros en la colección libros por su similitud con el título y retorna la colección de libros que tuvieron coincidencias.

```
5493858456@La Iliada@Homero@1782438932123@iliada.txt@1@
3432949392@La Divina Comedia@Dante Allghieri@2346342378950@0@
4320049400@Los miserables@Victor Hugo@2342678678564@1@
3342093029@La ceguera@Jose Saramago@23434567546433@0@
```

Fig. 1 Ejemplo de archivo

TEMA 3. (50 puntos)

Usted es parte del equipo de desarrollo de una empresa dedicada a la implementación de Juegos de Video y participará en la implementación del juego "Sudoku POO". Se le ha encargado la implementación de la interfaz gráfica acorde a los siguientes requerimientos:

- Crear la clase Celda la misma que contiene un valor y la posición en la cual ha sido ubicada dentro del tablero del Sudoku. Además, la clase debe dibujar el valor a través del método paintComponent. **(5 puntos)**
- Crear la clase Tablero, la cual está compuesta por una colección de Celdas ubicadas en forma de una matriz de 9x9. **(5 puntos)**
- Escriba el código para dibujar la GUI (SudokuFrame) que se encuentra en la Fig. 2: Sudoku POO. **(10 puntos)**
- Implemente la funcionalidad provista por el botón en la GUI, para insertar el número en las coordenadas indicadas por el usuario. Maneje la excepción SudokuException que se genera cuando un valor no puede ser colocado en una celda. **(10 puntos)**
- Al dar doble-click en una Celda, la interfaz debería de proveer una ventana de diálogo emergente para introducir el valor deseado sobre esa Celda. **(10 puntos)**
- Considere la interface Sudoku la misma que es implementada por la clase SudokuPOO que que mantiene la lógica del juego. Utilice una instancia de la clase SudokuPOO (atributo de la clase SudokuFrame) para cerciorarse de que los ingresos son posibles. Estas interfaces y clases, junto con SudokuException se encuentran dentro del paquete poo.sudoku. **(10 puntos)**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 8 | 1 | 5 | 6 | | | | | 4 |
| 6 | | | | 7 | 5 | | 8 | |
| | | | | 9 | | | | |
| 9 | | | | 4 | 1 | 7 | | |
| | 4 | | | | | | | 2 |
| | | 6 | 2 | 3 | | | | 8 |
| | | | | 5 | | | | |
| | 5 | | 9 | 1 | | | | 6 |
| 1 | | | | | 7 | 8 | 9 | 5 |

Fig. 2: Sudoku POO

```
interface Sudoku {
    // Recibe: Las coordenadas x,y del casillero.
    // Retorna: El número contenido en el casillero.
    // Lanza: Una excepción si el casillero está vacío.
    int getNumero(int x, int y) throws SudokuException;

    // Intenta insertar el número n en el tablero.
    // Recibe:
    // - El número a insertar.
    // - Las coordenadas x,y del casillero.
    // Retorna:
    // - true, insertable según las reglas del juego.
    // - false, si no se puede insertar
    boolean setNumero(int n, int x, int y);
}
```