

Certamen 3, Programación II

Prof. Rodrigo Olivares

Enero 4, 2017

Instrucciones:

- El puntaje máximo del certamen es 100%, siendo el 60% el mínimo requerido para aprobar.
- Responda la primera parte en la hoja que se le entrega (45 minutos). La segunda parte, responda segunda las indicaciones específicas.
- El certamen es **individual**. Cualquier intento de copia, será sancionado con nota **1,0**.

1. *18pts.* De las siguientes afirmaciones, encierre en un círculo la o las alternativas correctas. Pueden ser todas, algunas, una o ninguna de ellas.

i. (*3pts.*) Para construir una hebra se requiere:

- (a) Extender de una clase Thread.
- (b) Implementar una interfaz Thread.
- (c) Extender de una clase Run.
- (d) Implementar una interfaz Run.
- (e) Utilizar el método sleep.

ii. (*3pts.*) Para una hebra se debe:

- (a) Iniciar con el método run.
- (b) Iniciar con el método start.
- (c) Sobreescibir el método run.
- (d) Sobreescibir el método start.
- (e) Instanciar la hebra.

iii. (*4pts.*) En el ciclo de vida de una hebra, el estado:

- (a) New: crea, pero no inicializa la hebra.
- (b) Runnable: ejecuta la hebra, con tiempo CPU asignado.
- (c) Blocked: se ejecuta, sin importar estados internos.
- (d) Dead: es invocado generalmente por el método stop.
- (e) Yield: verifica el rendimiento del estado Runnable.

iv. (*3pts.*) Respecto a las interfaces gráfica en Java:

- (a) Swing sustituye a AWT.
- (b) AWT sustituye a Swing.
- (c) AWT se apoya en Swing.
- (d) AWT incorpora los JComponents.
- (e) Swing proporciona los ActionEvent.

v. (*3pts.*) Referente a JFrame:

- (a) Habitualmente se usa para crear la ventana principal.
- (b) Su método getContentPane() obtiene el panel principal.
- (c) Su método add() permite agregar componentes al panel.
- (d) Su método size() permite dimensionar la ventana.
- (e) Todas las anteriores

vi. (*4pts.*) Para las acciones desde un botón, se requiere:

- (a) Crear una clase que implemente un ActionEvent.
- (b) Crear una clase que implemente un ActionListener.
- (c) Sobreescibir el método actionPerformed(ActionEvent)
- (d) Sobreescibir el método actionPerformed(ActionEvent)
- (e) Agregar la instancia de la clase oyente, al botón.

2. 12pts. Considere el siguiente código.

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

public class View extends JFrame {

    private JTextField id = new JTextField(32), nombre = new JTextField(32), apellido =
        new JTextField(32);
    private JButton boton = new JButton("Validar");
    private JPanel panelTitulo = new JPanel(), panelBtn = new JPanel(), panelForm = new
        JPanel();

    public View() {
        super("View");
    }

    public void inicio() {
        setLayout(new BorderLayout());

        panelTitulo.setLayout(new FlowLayout(FlowLayout.CENTER, 10, 10));
        panelTitulo.add(new JLabel("Registro de datos"));
        add(panelTitulo, BorderLayout.NORTH);

        panelForm.setLayout(new FlowLayout(FlowLayout.LEFT, 0, 0));
        panelForm.add(new JLabel("ID Persona:"));
        panelForm.add(id);
        panelForm.add(new JLabel("Nombre:   "));
        panelForm.add(nombre);
        panelForm.add(new JLabel("Apellidos:  "));
        panelForm.add(apellido);
        add(panelForm, BorderLayout.CENTER);

        panelBtn.setLayout(new FlowLayout(FlowLayout.CENTER, 10, 10));
        boton.addActionListener(new ProcessListener());
        panelBtn.add(boton);
        add(panelBtn, BorderLayout.SOUTH);

        setSize(480, 200);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setVisible(true);
    }

    class ProcessListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            if (id.getText().isEmpty()) {
                JOptionPane.showMessageDialog(null, "ID Vacio", "Error", JOptionPane.
                    ERROR_MESSAGE);
            }
            if (nombre.getText().isEmpty()) {
                JOptionPane.showMessageDialog(null, "Nombre Vacio", "Error",
                    JOptionPane.ERROR_MESSAGE);
            }
            if (apellido.getText().isEmpty()) {
                JOptionPane.showMessageDialog(null, "Apellido Vacio", "Error",
                    JOptionPane.ERROR_MESSAGE);
            }
        }
    }

    public static void main(String[] args) {
        View v = new View();
        v.inicio();
    }
}
```

Describe y diagrama la o las tareas que pueden realizarse con esta aplicación.

3. *30pts.* Una empresa de autobuses le ha solicitado a Ud, como alumno de la Escuela de Ingeniería Civil Informática que desarrolle una simulación del proceso de subida y bajada de pasajeros. Los requisitos que le impone la empresa son:

- La simulación se realiza sólo considerando un autobs
- La capacidad máxima de un autobús es 45 pasajeros (sólo se consideran pasajeros sentados).
- Si el autobús está vacío, no puede detenerse para bajar pasajeros.
- Si el auto está lleno no puede detenerse para subir pasajeros.
- La cantidad de pasajeros que suben y bajan es aleatoria entre 1 y 10.
- La frecuencia de subida y bajada de pasajeros es aleatoria en 1 y 5 segundos.

Condiciones de entrega:

- **Debe compilar.**
- Debe considerar la creación de al menos dos hebras para los proceso y un recurso compartido.
- Subir al aula virtual **SÓLO** los archivos ***.java**, eliminado de ante mano la instrucción **package**, en un comprimido ApellidoPaternoNombreC3P3.zip. El no cumplimiento del formato será penalizado con 10 punto de descuento.
- Subir a la hora que se les indica en el certamen. El no cumplimiento con la hora de entrega, será penalizado con 1 punto de descuento por cada minuto de retraso.

4. *40pts.* Construya una aplicación en Java que permita, a través en una interfaz gráfica, realizar la transformación de código morse a código ASCII.

El alfabeto morse es el siguiente:

Signo	Código	Signo	Código	Signo	Código
A	.-	N	-.	0	- - - - -
B	-...	O	- - -	1	.. - - -
C	-.-.	P	.- -.	2	... - -
D	-.	Q	- -. -	3 -
E	.	R	..-	4-
F	..-.	S	...	5
G	- - .	T	-	6	-.....
H	U	..-	7	-
I	..	V-	8	- - - ...
J	.- - -	W	.- -	9	- - - - .
K	-. -	X	-. -.		
L	..-.	Y	-. - -		
M	- -	Z	- -.		

El código ASCII es el siguiente:

Signo	Código	Signo	Código	Signo	Código
0	0	C	C	O	O
1	1	D	D	P	P
2	2	E	E	Q	Q
3	3	F	F	R	R
4	4	G	G	S	S
5	5	H	H	T	T
6	6	I	I	U	U
7	7	J	J	V	V
8	8	K	K	W	W
9	9	L	L	X	X
A	A	M	M	Y	Y
B	B	N	N	Z	Z

Restricciones:

- Sólo se deben transformar palabras que contengan letras mayúsculas y números: A-Z y 0-9.
- Debe utilizar el paradigma de orientación a objetos.

Condiciones de entrega:

- **Debe compilar.**
- Subir al aula virtual **SÓLO** los archivos ***.java**, eliminado de ante mano la instrucción **package**, en un comprimido ApellidoPaternoNombreC3P4.zip. El no cumplimiento del formato será penalizado con 10 punto de descuento.
- Subir a la hora que se les indica en el certamen. El no cumplimiento con la hora de entrega, será penalizado con 1 punto de descuento por cada minuto de retraso.