

Certamen 3, Programación II

Prof. Rodrigo Olivares

Ayud. Juan Tapia

Agosto 30, 2016

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. La segunda parte, responda segunda las indicaciones específicas.
 - El certamen es **individual**. Cualquier intento de copia, será sancionado con nota **1,0**.
1. *30pts.* De las siguientes afirmaciones, encierre en un círculo la o las alternativas correctas. Pueden ser todas, algunas, una o ninguna [60 minutos].
- i. 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. Para una hebra o hilo se debe:
 - (a) Iniciar con el método run.
 - (b) Iniciar con el método join.
 - (c) Sobreescibir el método run.
 - (d) Sobreescibir el método start.
 - (e) Instanciar la hebra.
 - iii. En el ciclo de vida de una hebra, el estado:
 - (a) New crea e inicializa la hebra.
 - (b) Runnable ejecuta la hebra, si hay 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. Un recurso compartido:
 - (a) Puede ser una clase.
 - (b) Puede ser un objeto de la clase.
 - (c) Puede ser un atributo de la clase.
 - (d) Siempre debe estar sincronizado.
 - (e) Debe ser declarado como private o protected.
 - v. 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.
 - vi. 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
 - vii. Para realizar 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.
 - viii. Respecto a JDBC :
 - (a) Es un archivo ejecutable que instala clases e interfaces Java.
 - (b) Son interfaces que no dependen del SGBD.
 - (c) No se requiere de sentencias SQL.
 - (d) Es transparente al desarrollador.
 - (e) Implementan las clases e interfaces de java.sql.
 - ix. Su especificación incluye :
 - (a) Driver
 - (b) DriverManager
 - (c) Statement
 - (d) CallableStatement
 - (e) ResultSetStatement
 - x. Con JDBC, para registrar un controlador :
 - (a) Se utiliza el método `Class.forName("...")`
 - (b) Se utiliza el método `Class.setName("...")`
 - (c) Se utiliza el método `Class.forName("...")`
 - (d) Se utiliza el método `Class.getConnection("...")`
 - (e) Se utiliza el método `Class.forNameConnection("...")`

2. 70pts. Construya una aplicación en Java que muestre en una interfaz gráfica de usuario el funcionamiento de dos hebras: **ThreadToMorse** y **ThreadToASCII**. Estas hebras deben realizar una conversión de manera concurrente a una palabra generada por la clase RandomWord.

```
import java.util.Random;

public class RandomWord {

    public static String generate() {
        String randomStrings;
        Random random = new Random();
        char[] word = new char[random.nextInt(8) + 3];

        for(int j = 0; j < word.length; j++) {
            word[j] = (char)('a' + random.nextInt(26));
        }

        return new String(word);
    }
}
```

Para mostrar los resultados debe utilizar la siguiente estructura:

```
import java.awt.BorderLayout;
import java.awt.FlowLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

public class View extends JFrame {

    public JTextArea out;
    private JButton processBtn;
    private JPanel panelTitle, panelOut, panelBtn;

    public View() {
        super("GUI + Threads");
    }

    public void start() {
        out = new JTextArea(80, 30);
        processBtn = new JButton("Inicio");
        panelTitle = new JPanel();
        panelOut = new JPanel();
        panelBtn = new JPanel();

        setLayout(new BorderLayout());

        panelTitle.setLayout(new FlowLayout(FlowLayout.CENTER, 10, 10));
        panelTitle.add(new JLabel("Concurrencia"));
        add(panelTitle, BorderLayout.NORTH);

        panelOut.setLayout(new FlowLayout(FlowLayout.CENTER, 0, 0));
        panelOut.add(out);
        add(panelOut, BorderLayout.CENTER);

        panelBtn.setLayout(new FlowLayout(FlowLayout.CENTER, 10, 10));
        panelBtn.add(processBtn);
        add(panelBtn, BorderLayout.SOUTH);

        setSize(380, 400);

        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
```

Debe mejorar la vista de maneta tal que los mensajes codificados, tanto a *morse* como a *ascii* se pueden visualizar correctamente, por ejemplo agregando un JScrollPane.

La clase principal debe ser la siguiente

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

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 las clases entregadas.
- 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 ApellidoMNombreC3.zip. El no cumplimiento del formato
- 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.