

QUINTA SESION
Programación con Lenguajes Orientados a Objetos
Prof. Gilberto Manzano Peñaloza

Alumno: _____ Fecha: _____

Indicaciones: Diseña una solución en Java para los siguientes problemas recursivos:

- 5.1. Cuando queremos indicar productos de factores iguales, generalmente usamos la notación exponencial. En la notación exponencial la base es el factor que debe multiplicarse por sí mismo tantas veces como lo indica el exponente. Por ejemplo 8^5 se pueden expresar por $8 \times 8 \times 8 \times 8 \times 8$ de otro modo 8 a la 5ª potencia.

Cuando la base de una potencia es un número positivo, el resultado siempre es positivo. Si la base es negativa, el signo del resultado depende del exponente, si el exponente es un número par, el resultado es positivo, si es impar el resultado es negativo. Si el exponente es uno, el resultado es igual a la base de la potencia. Si el exponente es cero, el resultado de la potencia siempre es igual a 1.

A) Realiza la formulación recursiva de potencia.

- 5.2. El problema de las Torres de Hanoi es un problema clásico de recursividad.

Problema: Tenemos tres torres: A, B y C y un conjunto de 5 discos, todos de distintos tamaños. El juego comienza con todos los discos colocados en la torre A de tal forma que ninguno de ellos cae sobre uno de menor tamaño; es decir, están apilados, uno sobre otro, con el disco más grande hasta abajo, encima de este el siguiente en tamaño y así sucesivamente. El propósito del juego es apilar los cinco discos, en el mismo orden, pero en la torre C. Durante la solución puede colocar los discos en cualquier torre, pero debe apegarse a las siguientes reglas:

- Sólo puede mover el disco superior de cualquiera de las torres.
- Un disco más grande nunca puede estar encima de uno más pequeño.

A) Intenta resolver el juego manualmente.

B) Explica en forma de pseudocódigo los pasos que seguiste para resolver el enigma.

C) Explica que pasos son recursivos.

D) Diseña una función recursiva nombrada Hanoi para este problema.

- 5.3. Diseña una solución recursiva en el que dadas dos series, una de letras y otra de números, las dos de igual longitud, imprima los posibles pares donde el primer elemento es una letra y el segundo un número.

Ejemplo: Dada una serie A, B, C y otra 1, 2, 8
 Imprima A1, A2, A8
 B1, B2, B8
 C1, C2, C8

QUINTA SESION

Programación con Lenguajes Orientados a Objetos

Solución al problema de las torres de hanoi:

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

public class Hanoi extends JApplet implements ActionListener {
    JTextArea resultado;
    JTextField textoEntrada;
    String salida = "";
    int contador=0;
    // inicializa el applet
    public void init() {
        JLabel lblTitulo, lblEntrada, lblSalida;
        JButton btnProcesar;
        Container contString = getContentPane();
        contString.setLayout(new FlowLayout());
        lblTitulo = new JLabel ("    Las Torres de Hanoi                ");
        contString.add(lblTitulo);
        lblEntrada = new JLabel("Digite la cantidad de discos");
        contString.add(lblEntrada);
        textoEntrada = new JTextField(4);
        contString.add(textoEntrada);
        textoEntrada.addActionListener(this);
        btnProcesar = new JButton("Procesar");
        btnProcesar.addActionListener(this);
        contString.add(btnProcesar);
        lblSalida = new JLabel(" Resultado: ");
        contString.add(lblSalida);
        resultado = new JTextArea(15, 45);
        resultado.setEnabled(false);
        contString.add(new JScrollPane(resultado));
    }

    public void actionPerformed (ActionEvent e){
        //inicializa en 0 el contador y la salida
        contador=0;
        salida="";
        // Definición de los chars
        char l = 'l';
        char T = 'T';
        char D = 'D';
        //convertir lo que se escribe en textoEntrada a entero
        String Entrada = "", TiLi="";
        Entrada = textoEntrada.getText();
        int X = Integer.parseInt(Entrada);
```

QUINTA SESION
Programación con Lenguajes Orientados a Objetos

```
if (X==1){ TiLi=" ----Solicitaste "+X+" disco----\n\n";}
    else {TiLi=" ----Solicitaste "+X+" discos-----\n\n";}

//Control de Error, el número debe ser mayor que 0
if (X<=0) { resultado.setText(TiLi+
    "::ERROR:\nNecesito al menos un disco para trabajar");}
else resultado.setText(TiLi+Solve(X,I,T,D));

}

//Método Move, escribe el movimiento debido y suma al contador
public String Move(char I, char D){
    String mover= "Mover disco en "+I+" hasta "+D+"\n";
    contador++;
    return " movimiento "+contador+": "+mover;
}

//Método Solve, encargado de solucionar el problema
public String Solve (int X, char I, char T, char D){
//Necesita el momento en que llegue a X==0 pues usa recursividad
//hasta que el Stack cumple con X=0 que es cuando empieza a sacar
//los valores requeridos
    if (X==0)
        salida +="";
    else {
        Solve (X-1,I,D,T);
        salida += Move(I,D);
        Solve (X-1,T,I,D);
    }
return salida;
}

public Hanoi() {
    try {
        jblnit();
    }
    catch(Exception e) {
        e.printStackTrace();
    }
}

private void jblnit() throws Exception {
    this.getContentPane().setBackground(SystemColor.controlLtHighlight);
}
}
```