



TAREA 23

JUEGO DE TRES EN RAYA

- Crear un programa que muestre una grilla de 3x3, llenado con botones en blanco
- Cuando se cliquee el primer botón, que el texto del botón cambie a X, subsecuentemente al ir clickeando los botones se debería marcar un O, luego un X, alternadamente

```
3 import java.awt.Color;
4 import java.awt.GridLayout;
5 import java.awt.event.ActionEvent;
6 import java.awt.event.ActionListener;
7 import java.util.*;
8
9 import javax.swing.*;
10 import javax.swing.JFrame;
11
12 public class TresEnLineaFrame extends JFrame {
13     private final int ANCHO = 400;
14     private final int ALTO = 400;
15
16     private JButton c1,c2,c3,c4,c5,c6,c7,c8,c9;
17     private HashMap<JButton,Integer> botones = new HashMap<JButton,Integer>();
18
19     public TresEnLineaFrame() {
20         setTitle("Tres en Linea");
21         setSize(ANCHO,ALTO);
22         setLayout(new GridLayout(3,3));
23         setDefaultCloseOperation(EXIT_ON_CLOSE);
24         createContents();
25         setVisible(true);
26
27     }
```

```
28 private void createContents() {
29     Listener listener = new Listener();
30
31     c1 = new JButton();
32     c2 = new JButton();
33     c3 = new JButton();
34     c4 = new JButton();
35     c5 = new JButton();
36     c6 = new JButton();
37     c7 = new JButton();
38     c8 = new JButton();
39     c9 = new JButton();
40     inicalizarBotones();
41
42     add(c1);
43     add(c2);
44     add(c3);
45     add(c4);
46     add(c5);
47     add(c6);
48     add(c7);
49     add(c8);
50     add(c9);
51
52     c1.addActionListener(listener);
53     c2.addActionListener(listener);
54     c3.addActionListener(listener);
55     c4.addActionListener(listener);
56     c5.addActionListener(listener);
57     c6.addActionListener(listener);
58     c7.addActionListener(listener);
59     c8.addActionListener(listener);
60     c9.addActionListener(listener);
61
62 }
63 public void inicalizarBotones() {
64     botones.put(c1,0);
65     botones.put(c2,1);
66     botones.put(c3,2);
67     botones.put(c4,10);
68     botones.put(c5,11);
69     botones.put(c6,12);
70     botones.put(c7,20);
71     botones.put(c8,21);
72     botones.put(c9,22);
73 }
```

```

74 private class Listener implements ActionListener {
75     private String [][] casillas = new String [3][3];
76     private String [] valores = {"X","O"} ;
77     private int valor = 0;
78     private ArrayList<Integer> posiciones;
79     private String ganador=null;
80
81     public void actionPerformed(ActionEvent e) {
82         JButton c = (JButton)e.getSource();
83         llenarCasilla(c);
84
85         if(casilleroLlenos()>=5) {
86             posiciones = verificarTresEnLinea(c);
87             if(posiciones.size()==3) {
88                 ganador = c.getText();
89                 if(ganador.equals("X")) {
90                     pintarTresEnLinea(posiciones,Color.YELLOW);
91                 }
92                 else {
93                     pintarTresEnLinea(posiciones,Color.GREEN);
94                 }
95                 JOptionPane.showMessageDialog(null, "Juego Terminado GANA "+ganador);
96                 //salir de la ventana
97                 System.exit(0);
98             }
99         }
100         if(casilleroLlenos()==9 && ganador==null) {
101             JOptionPane.showMessageDialog(null, "Juego Terminado EMPATE ");
102             // salir de la ventana
103             System.exit(0);
104         }
105     }
106
107     public void llenarCasilla(JButton button) {
108         int fila,col;
109         fila = botones.get(button)/10;
110         col = botones.get(button)%10;
111         if(casillas[fila][col]==null) {
112             casillas[fila][col]=valores[valor];
113             button.setText(valores[valor]);
114             valor= (valor+1)%2;
115         }
116         else {
117             JOptionPane.showMessageDialog(null, "el casillero esta ocupado");

```

```

120 public ArrayList<Integer> verificarTresEnLinea(JButton button) {
121
122     int fila = botones.get(button)/10;
123     int col = botones.get(button)%10;
124     ArrayList<Integer> posiciones;
125
126     posiciones = verificarFilaColumna(fila,col);
127     if(posiciones.size()<2){
128         if((fila ==0 & col==0) || (fila==2 &col==2)) {
129             posiciones = verificarDiagonal(button,1);
130         }
131         else if((fila ==0 & col==2) || (fila==2 &col==0)) {
132             posiciones = verificarDiagonal(button,2);
133         }
134         else if(fila ==1 & col==1) {
135             posiciones = verificarCentro(button);
136         }
137     }
138     posiciones.add(botones.get(button));
139
140     return posiciones;
141 }
142 public ArrayList<Integer> verificarFilaColumna(int fila,int col) {
143
144     ArrayList<Integer> posiciones = new ArrayList<Integer>();
145     for(int i=1;i<=2;i++) {
146         int c=(col+i)%3;
147         if((casillas[fila][c]!=null) && (casillas[fila][c].equals(casillas[fila][col]))) {
148             posiciones.add(fila*10+c);
149         }
150     }
151     if(posiciones.size()<2) {
152         posiciones.clear();//
153         for(int i=1;i<=2;i++) {
154             int f=(fila+i)%3;
155             if((casillas[f][col]!=null) && (casillas[f][col].equals(casillas[fila][col]))) {
156                 posiciones.add(f*10+col);
157             }
158         }
159     }
160     return posiciones;
161 }

```

```

162 public ArrayList<Integer> verificarDiagonal(JButton button,int op) {
163     int fila = botones.get(button)/10;
164     int col = botones.get(button)%10;
165     ArrayList<Integer> posiciones = new ArrayList<Integer>();
166     int c=col;
167     for(int i=1;i<=2;i++) {
168         int f=(fila+i)%3;
169         c=(c+op)%3; //op 1= izq 2 =derecha
170         if((casillas[f][c]!=null) && (casillas[f][c].equals(casillas[fila][col]))) {
171             posiciones.add(f*10+c);
172         }
173     }
174     return posiciones;
175 }
176 public ArrayList<Integer> verificarCentro(JButton button) {
177     ArrayList<Integer> posiciones = new ArrayList<Integer>();
178     posiciones = verificarDiagonal(button,1);
179     if(posiciones.size()<3) {
180         posiciones.clear();
181         posiciones = verificarDiagonal(button,2);
182     }
183     return posiciones;
184 }
185 public int casilleroLlenos() {
186     int llenos = 0;
187     for(int i=0;i<casillas.length;i++) {
188         for(int j=0;j<casillas[i].length;j++) {
189             if(casillas[i][j]!=null) {
190                 llenos++;
191             }
192         }
193     }
194     return llenos;
195 }
196 public void pintarTresEnLinea(ArrayList<Integer>posiciones,Color c) {
197     for(int pos:posiciones) {
198         for(JButton b: botones.keySet()) {
199             if(botones.get(b)==pos) {
200                 b.setBackground(c);
201             }
202         }
203     }
204 }
205 }

```

```

207 public static void main(String[] args) {
208     TresEnLineaFrame michi = new TresEnLineaFrame ();
209
210 }
211
212 }
---
```

