

```

package simuVIH;

/*
 * @author Brayan_Ramirez_Benitez
 */
import javax.swing.*.*;
import java.awt.*.*;

public class SimuVIH {

    //Partes para el Frame
    JFrame frame;
    JPanel estado;
    JButton[][] cuadritos; //Botones que actuaran como celulas

    int[][] contenedor;
    int[][] estadoActual;
    int[][] estadoSiguiente;
    int[][] auxiliar;

    int filas = 50; //Establecera el tamaño del Panel y el tiempo de espera
    int columnas = 50;
    int tiempo = 1300;

    public SimuVIH(){} //Soy un constructor :D

    private void CreaPanel(){} // En este metodo creamos los botones del Panel y establecemos el
    tamaño

    cuadritos = new JButton[filas][columnas];

    estado = new JPanel();

```

```
estado.setLayout(new GridLayout(filas,columnas));
```

```
for(int i = 0 ; i < filas ; i++){  
    for(int j = 0 ; j < columnas ; j++){  
        JButton cuadro = new JButton();  
        String id = String.valueOf(i) + "-" + String.valueOf(j);  
        cuadro.setName(id);  
        cuadro.setSize(10, 10);  
        cuadro.setToolTipText(id);  
        cuadritos[i][j] = cuadro;  
    }  
}  
for(int i = 0 ; i < filas ; i++){  
    for(int j = 0 ; j < columnas ; j++){  
        estado.add(cuadritos[i][j]);  
    }  
}  
}
```

private void Ventana(){//En este metodo establecemos la ventana para el Panel y le colocamos un titulo

```
frame = new JFrame("Simulacion VIH");  
frame.setSize(900, 900);  
frame.add(estado);  
frame.setVisible(true);  
frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);  
}
```

```
private void Simulacion(){//En este metodo ejecutaremos la simulacion
```

```
    contenedor = new int[3][3];  
    estadoActual = new int[filas][columnas];  
    estadoSiguiente = new int[filas][columnas];  
    auxiliar = new int[filas][columnas];
```

```
    IniciaCuadros(estadoActual);  
    RellenaCuadros();  
    IniciaColores();  
    ColorCuadros(estadoActual);
```

```
do{//Iniciamos un bucle para que se ejecute indefinidamente
```

```
    for(int i= 0 ; i < filas -2 ; i++){  
        for(int j = 0 ; j < columnas -2 ; j++){  
            int c1 = estadoActual[(i)][(j)];  
            int c2 = estadoActual[(i)][(j+1)];  
            int c3 = estadoActual[(i)][(j+2)];  
            int c4 = estadoActual[(i+1)][(j)];  
            int c5 = estadoActual[(i+1)][(j+1)];  
            int c6 = estadoActual[(i+1)][(j+2)];  
            int c7 = estadoActual[(i+2)][(j)];  
            int c8 = estadoActual[(i+2)][(j+1)];  
            int c9 = estadoActual[(i+2)][(j+2)];
```

```
            contenedor[0][0] = c1;  
            contenedor[0][1] = c2;  
            contenedor[0][2] = c3;  
            contenedor[1][0] = c4;
```

```
contenedor[1][1] = c5;
```

```
contenedor[1][2] = c6;
```

```
contenedor[2][0] = c7;
```

```
contenedor[2][1] = c8;
```

```
contenedor[2][2] = c9;
```

```
int contador = 0, contador1 = 0, k, l, p;
```

```
for(k = 0 ; k < 3 ; k++){
```

```
    for( l= 0 ; l < 3 ; l++){
```

```
        if( !(k == 1 && l == 1)){
```

```
            if(contenedor[k][l] == 1){//cuenta las A alrededor de un boton
```

```
                contador++;
```

```
        }
```

```
        else if(contenedor[k][l] == 2){//cuenta las B alrededor de un boton
```

```
            contador1++;
```

```
        }
```

```
    }
```

```
}
```

```
//Una celula se infecta de A si tiene al menos 1
```

```
if(contador >= 1 && contenedor[1][1] == 0){
```

```
    estadoSiguiente[(i+1)][(j+1)] = 1;
```

```
//Una celula se infecta de B si tiene al menos 4
```

```
else if(contador1 == 4 && contenedor[1][1] == 0){
```

```
    estadoSiguiente[(i+1)][(j+1)] = 2;
```

```
//Una celula A se propaga y convierte en B
```

```
else if(contenedor[1][1] == 1){
```

```
    estadoSiguiente[(i+1)][(j+1)] = 2;
```

```
}
```

```

        else if(contenedor[1][1] == 2){ //Una celula B se propaga y Muere
            estadoSiguiente[(i+1)][(j+1)] = 3;
        }

        else if(contenedor[1][1] == 3){ //Una probabilidad del 0.99 de ser una nueva celula y
3*10^-3 de estar infectada con A
            p = (int) (Math.random()*325);
            if (p == 50){
                estadoSiguiente[(i+1)][(j+1)] = 1;
            }
        }
    }
}

auxiliar = estadoActual;
estadoActual = estadoSiguiente;
estadoSiguiente = auxiliar;

try{
    Thread.currentThread().sleep(tiempo);
}catch (InterruptedException ie){
    System.out.println("Exception " + ie.toString() );
}

ColorCuadros(estadoSiguiente);
IniciaCuadros(estadoSiguiente);
}
while(true);
}

```

private void ColorCuadros(int[][] Cuadros){// Este metodo asignara el valor de cada boton para una celula

```
for(int i = 0 ; i < filas ; i++){
    for(int j = 0; j < columnas ; j++){
        if(Cuadros[i][j] == 3){
            cuadritos[i][j].setBackground(Color.RED);//Muertas
        }
        if(Cuadros[i][j] == 2){
            cuadritos[i][j].setBackground(Color.GREEN);// infectadas B
        }
        if(Cuadros[i][j] == 1){
            cuadritos[i][j].setBackground(Color.YELLOW);//infectadas A
        }
        if(Cuadros[i][j] == 0){
            cuadritos[i][j].setBackground(Color.BLUE);//sanas
        }
    }
}
```

private void IniciaCuadros(int[][] Cuadros){//Este metodo nos permitira asignarle el valor de una celula sana a cada boton

```
for(int i = 0 ; i < filas ; i++){
    for(int j= 0 ; j < columnas ; j++){
        Cuadros[i][j] = 0;//Inicia sana
    }
}
```

private void IniciaColores(){//Este metodo nos permitira asignarle el color a cada boton como una celula sana (Azul)

```
for(int i = 0 ; i < filas ; i++){
    for(int j = 0; j < columnas ; j++){
        cuadritos[i][j].setBackground(Color.BLUE);
    }
}
}
```

private void RellenaCuadros(){//Este metodo nos permitira asignarle el color a las celulas infectadas

```
for(int j= 0 ; j < 10 ; j++){//900*0.05 = 45
    int y = (int)Math.floor(Math.random()*(0-filas+1)+filas);
    int x = (int)Math.floor(Math.random()*(0-columnas+1)+columnas);
    estadoActual[y][x] = 1;
}
}
```

public static void main(String [] Game){

```
    SimuVIH simu = new SimuVIH();
    simu.CreaPanel();//Llamamos al metodo para crear el Panel
    simu.Ventana();//Creamos la ventana para el Panel
    simu.Simulacion();//Ejecutamos la simulacion
}
```

}