



PRÁCTICA 2

PROGRAMACIÓN



El juego de la vida

<p>Realizado por Ignacio Peribáñez Belanche Curso 2020/2021</p>

CLASE MAIN

1- Método main:

```
163● public static void main(String[] args) {
164    Scanner entrada = new Scanner(System.in);
165    System.out.println("¡Bienvenidos al juego de la vida!");
166    System.out.print(
167        "-----\n");
168    while (true) {
169        System.out.println("");
170        System.out.println("0.- Apagar programa");
171        System.out.println("1.- Generar tablero aleatorio");
172        System.out.println("2.- Pruebas");
173        System.out.print("\n¿Que opcion deseas ejecutar?: ");
174        int numero = entrada.nextInt();
175        System.out.println("");
176        if (numero > 2 || numero < 0) { // Mostramos el error en caso de introducir una opcion inexistente
177            System.out.println("Error, elija una de las opciones validas");
178        } else if (numero == 0) { // Paramos el programa al introducir 0
179            System.out.print("Has apagado el programa");
180            break;
181        } else if (numero == 1) { // Llamamos al metodo aleatorio() si el numero es 1
182            aleatorio();
183        } else if (numero == 2) { // Llamamos al main() de la clase Prueba si el numero es 2
184            Prueba.main(args);
185        }
186    }
187 }
188 }
```

El método main se encarga de mostrar un menú con el que movernos por el programa.

2- Método aleatorio:

```
141● public static void aleatorio() {
142    int filas = -1;
143    int columnas = -1;
144    int generaciones = -1;
145    Scanner entrada = new Scanner(System.in);
146    while (filas < 0) { // Bucle infinito hasta que el numero sea positivo
147        System.out.print("Dame un numero de filas: ");
148        filas = entrada.nextInt();
149    }
150    while (columnas < 0) { // Bucle infinito hasta que el numero sea positivo
151        System.out.print("Dame un numero de columnas: ");
152        columnas = entrada.nextInt();
153    }
154    while (generaciones < 0) { // Bucle infinito hasta que el numero sea positivo
155        System.out.print("Introduce un número de generaciones: ");
156        generaciones = entrada.nextInt();
157    }
158    generaciones(generaciones, filas, columnas);
159    System.out.print(
160        "\n-----\n");
161 }
```

Este metodo se encarga de pedir un numero de filas, columnas y generaciones al usuario, y llamar al metodo generaciones() enviando los datos introucidos por el usuario.

3- Método tripleta:

```
12● public static Tripleta tripleta(String[][] t,int i, int f, int c, int nuevas) {
13    int vivas = 0;
14    for (int z = 0; z < f; z++) { // Bucle donde remplazaremos la tabla t por la tabla t2 despues de cada generacion
15        for (int j = 0; j < c; j++) {
16            if (t[z][j].equals("**")) {
17                vivas++;
18            }
19        }
20    }
21    Tripleta tri = new Tripleta(i, vivas, nuevas);
22    return tri;
23 }
```

En este método primero comprobamos la cantidad de células vivas que hay en esta generación, creamos un objeto tipo Tripleta y lo devolvemos.

2- Método generaciones:

(Este método es lo dividiremos en partes debido a su tamaño.)

```
29 public static void generaciones(int g, int f, int c) {
30     ArrayList<Tripleta> l = new ArrayList<Tripleta>();
31     String t[][] = new String[f][c];
32     String t2[][] = new String[f][c]; // Creamos una tabla identica a la tabla "t", donde escribiremos dependiendo de las condiciones.
33     for (int i = 0; i < f; i++) { // Creamos un bucle que crea la generacion 0 y escribe en la tabla "t2" para que no tenga valor null.
34         for (int x = 0; x < c; x++) {
35             int numero = (int) Math.floor(Math.random() * 9);
36             if (numero == 0 || numero == 1) {
37                 t[i][x] = "***";
38             } else {
39                 t[i][x] = " ";
40             }
41             t2[i][x] = t[i][x];
42         }
43     }
44 }
45 System.out.println("\nGeneración 0");
46 for (int z = 0; z < f; z++) { // Creamos un bucle que se encarga de mostrar la generacion 0
47     System.out.println();
48     for (int x = 0; x < c; x++) {
49         System.out.print(t2[z][x]);
50         System.out.print(" ");
51     }
52 }
```

Primero creamos un ArrayList de tipo Tripleta y 2 tablas multidimensionales (la longitud de dichas tablas las obtenemos del método aleatorio())

Crearemos la generación 0, rellenando las 2 tablas con los mismos datos y la mostraremos por pantalla.

```
54 l.add(tripleta(t, 0, f, c, 0));
55 for (int i = 1; i < g; i++) { // Creamos un bucle que crea cada generacion (despues de la generacion 0).
56     System.out.println("\nGeneración " + i);
57     /**
58      * Bucle donde reemplazaremos la tabla t por la tabla t2 despues de cada generacion, esta programado de manera
59      * que la primera vez que se ejecute el bucle, este no cambie nada.
60      */
61     for (int z = 0; z < f; z++) {
62         for (int j = 0; j < c; j++) {
63             t[z][j] = t2[z][j];
64         }
65     }
66     int nuevas = 0;
67     for (int j = 0; j < f; j++) { // Creamos un bucle que se encarga de determinar el estado de cada celula
68         for (int x = 0; x < c; x++) {
69             int contador = 0;
70             try {
71                 if (t[j - 1][x].equals("***")) {
72                     contador++;
73                 }
74                 if (t[j - 1][x - 1].equals("***")) {
75                     contador++;
76                 }
77                 if (t[j - 1][x + 1].equals("***")) {
78                     contador++;
79                 }
80                 if (t[j][x - 1].equals("***")) {
81                     contador++;
82                 }
83                 if (t[j][x + 1].equals("***")) {
84                     contador++;
85                 }
86                 if (t[j + 1][x - 1].equals("***")) {
87                     contador++;
88                 }
89                 if (t[j + 1][x].equals("***")) {
90                     contador++;
91                 }
92                 if (t[j + 1][x + 1].equals("***")) {
93                     contador++;
94                 }
95             } catch (ArrayIndexOutOfBoundsException e) {}
96         }
97     }
98 }
```

```

96      /*
97      * El "ArrayIndexOutOfBoundsException" se encarga de evitar
98      * el error al comparar una celda con otra celda inexistente.
99      */
100     if (t[j][x].equals("")) {
101         if (contador < 2) {
102             t2[j][x] = " ";
103         } else if (contador > 3) {
104             t2[j][x] = " ";
105         }
106     } else {
107         if (contador == 3) {
108             t2[j][x] = "*";
109             nuevas ++;
110         }
111     }
112 }
113 }
114 }
115 }
116 for (int z = 0; z < f; z++) { // Creamos un bucle que se encarga de mostrar la tabla despues de las condiciones
117     System.out.println();
118     for (int x = 0; x < c; x++) {
119         System.out.print(t2[z][x]);
120         System.out.print(" ");
121     }
122 }
123 Tripletta triple = tripletta(t2, i, f, c, nuevas);
124 System.out.println("\n\nSobreviven " + triple.getVivas() + " células");
125 l.add(triple); // Añadimos la tripletta correspondiente a cada generacion a partir de la 1
126 if (triple.getVivas() == 0) { // Si no hay celulas vivas ejecutamos el break.
127     System.out.print("\nColonia extinguida\n");
128     break;
129 }
130 }
131 for(int i = 0; i < l.size(); i++) { // Bucle para mostrar las tripletas de cada generacion
132     System.out.println("\n" + l.get(i));
133 }
134 }

```

Añadimos la generación 0 al ArrayList y comenzamos a crear las generaciones que nos ha pedido el usuario, restándole 1 ya que la generación 0 ya esta creada fuera del bucle, para las nuevas generaciones, en estas generaciones actuaran las condiciones.

Por cada generación crearemos un objeto tipo Tripletta y lo añadiremos al ArrayList.

Si la tripletta correspondiente a esa generación no tiene células vivas el bucle se rompe y mostramos el ArrayList.

CLASE PRUEBA

1- Método main:

Muestra un menú, y dependiendo de la opción elegida mandara unos parámetros al método prueba().

2- Método prueba:

Se encarga de crear las generaciones y las tablas, dependiendo de la opción elegida, rellenara las tablas de una forma u otra.

CLASE TRIPLETA

Es el constructor de el objeto tipo Tripletta.