



EJERCICIO THREADS 7

Implemente una simulación de la fábula que cuenta la carrera entre la liebre y la tortuga. Para hacerlo más interesante la carrera será cuesta arriba por una pista resbaladiza, de modo que a veces podrán resbalar y retroceder algunas posiciones.

Habrà un thread que implementará la tortuga y otro la liebre. No debe haber intervención del usuario. Debe ser un bucle hasta que uno de los dos gana. Entonces se saca por pantalla que thread a acabado antes. Cada thread guarda un valor llamado casilla, que cuando llega a 100 acaba.

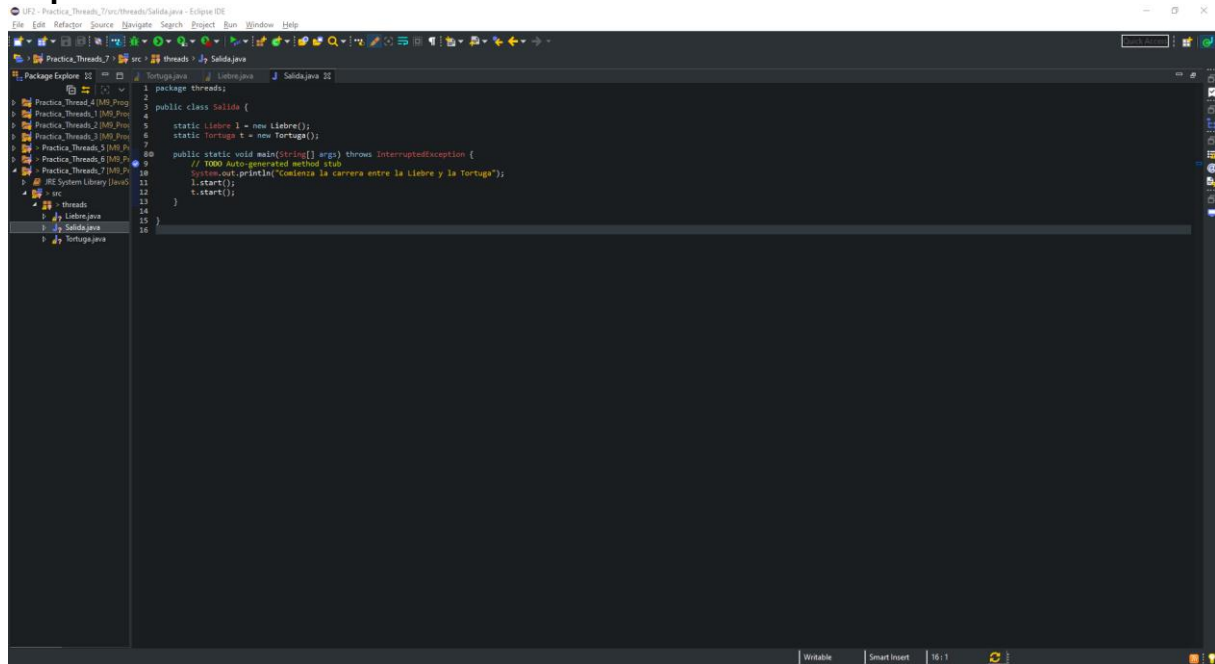
Cada uno se suspenderá durante un segundo y luego evaluará lo que ha pasado según unas probabilidades:

ANIMAL	Suceso	Probabilidad	Movimiento
Tortuga	Avance rápido	50%	3 casillas adelante
	Resbalón	20%	6 casillas atrás
	Avance lento	30%	1 casilla adelante
Liebre	Duerme	20%	No se mueve
	Gran salto	20%	9 casillas adelante
	Resbalón grande	10%	12 casillas atrás
	Pequeño salto	30%	1 casilla adelante
	Resbalón pequeño	20%	2 casillas atrás

Calcule la probabilidad con random, de 1 a 100 y determine con dicho número que ha hecho cada animal. Considere que hay 70 casillas, de la 1 a la 70, la 1 de salida y la 70 de llegada. Si resbala al principio vuelve a la 1, nunca por debajo. Tras cada segundo y después de calcular su nueva posición imprima las posiciones, y luego una letra T para la tortuga y una L para la liebre. Imprima al comienzo de la carrera un mensaje. Después de imprimir las líneas determine si alguno ha llegado a meta y ha ganado, imprimiendo un mensaje. Si ambos llegan a la vez declare un empate



Generalitat de Catalunya
Departament d'Educació
Institut d'Educació Secundària i Superior
d'Ensenyaments Professionals
Esteve Terradas i Illa
Departament d'Informàtica



// TORTUGA

```
package threads;
```

```
public class Tortuga extends Thread {
```

```
    public static int casilla = 1; // Casilla que avanzan la Tortuga
```

```
    private int maxCasilla = 0; // Maximo casillas que puede moverse para  
terminar la carrera
```

```
    private int avance_rapido = 3;
```

```
    private int resbalon = 6;
```

```
    private int avance_lento = 1;
```

```
    public boolean ganador = false;
```

```
    public void run() {
```

```
        /*
```

```
        * While para las 100 casillas si en 100 casillas no a sido capaz de  
        llegar a la meta no es valido.
```

```
        */
```

```
        while (maxCasilla != 100) {
```

```
        // ESPERAMOS UN SEGUNDO ENTRE CALCULO Y CALCULO DE LA NUEVA POSICION
```

```
        try {
```

```
            Thread.sleep(1000);
```

```
        } catch (InterruptedException e1) {
```

```
            // TODO Auto-generated catch block
```

```
            e1.printStackTrace();
```

```
        }
```

```
        int dados = (int) (Math.random() * 100) + 1;
```

```
        // RESBALON
```

```
        if (dados >= 1 && dados <= 20) {
```

```
            casilla = casilla - resbalon;
```

```
            if (casilla < 1) {
```



```
        casilla = 1;
    }

    // AVANCE LENTO
} else if (dados >= 21 && dados <= 50) {
    casilla = casilla + avance_lento;
    if (casilla > 70) {
        casilla = 70;
    }

    // AVANCE RAPIDO
} else if (dados >= 51 && dados <= 100) {
    casilla = casilla + avance_rapido;
    if (casilla > 70) {
        casilla = 70;
    }
}
System.out.println("Posicion T: " + casilla);

/*
 * Aqui lo que hago es que cuando llega la casilla 70 es que la
tortuga a llegado a la meta
 * entonces significa que a ganado. Paro el thread de la
tortuga para que no siga corriendo y compruebo si ella tambien a llegado a la
misma vez a la meta.
 * Si a llegado imprimo un empate si no a llegado imprimo que
la tortuga a ganado
 */
if (casilla == 70) {
    Salida.l.stop();

    try {
        // Me aseguro que el otro thread se a parado para
determinar el final de la carrera con el join
        Salida.l.join();
    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    if (Liebre.casilla == 70) {
        System.out.println("\n EMPATE");
    } else {
        System.out.println("\n La Tortuga a llegado a la
meta y ha ganado!");
    }
    break;
}
maxCasilla++;
}
}

}

//LIEBRE
package threads;
```



Generalitat de Catalunya
Departament d'Educació
Institut d'Educació Secundària i Superior
d'Ensenyaments Professionals
Esteve Terradas i Illa
Departament d'Informàtica

```
public class Liebre extends Thread {

    public static int casilla = 1; // Casilla que avanza la Tortuga
    private int maxCasilla = 0; // Maximo casillas que puede moverse para
terminar la carrera
    private int duerme = 0;
    private int gran_salto = 9;
    private int resbalon_grande = 12;
    private int pequeño_salto = 1;
    private int resbalon_pequeño = 2;
    public boolean ganador = false;

    public void run() {

        /*
         * While para las 100 casillas si en 100 casillas no a sido capaz de
llegar a la meta no es valido.
         */
        while (maxCasilla != 100) {
// ESPERAMOS UN SEGUNDO ENTRE CALCULO Y CALCULO DE LA NUEVA POSICION
try {
            Thread.sleep(1000);
        } catch (InterruptedException e1) {
// TODO Auto-generated catch block
e1.printStackTrace();
        }

        // Dados que tira la liebre para avanzar o retroceder casillas
        int dados = (int) (Math.random() * 100) + 1;
        // RESBALON GRANDE
        if (dados >= 1 && dados <= 10) {
            casilla = casilla - resbalon_grande;
            if (casilla < 1) {
                casilla = 1;
            }

            // DUERME
        } else if (dados >= 11 && dados <= 30) {
            casilla = casilla + duerme;
            if (casilla > 70) {
                casilla = 70;
            }

            // GRAN SALTO
        } else if (dados >= 31 && dados <= 50) {
            casilla = casilla + gran_salto;
            if (casilla > 70) {
                casilla = 70;
            }

            // RESBALON PEQUEÑO
        } else if (dados >= 51 && dados <= 70) {
            casilla = casilla - resbalon_pequeño;
            if (casilla < 1) {
                casilla = 1;
            }
        }
    }
}
```



Generalitat de Catalunya
Departament d'Educació
Institut d'Educació Secundària i Superior
d'Ensenyaments Professionals
Esteve Terradas i Illa
Departament d'Informàtica

```
    }  
  
    // PEQUEÑO SALTO  
} else if (dados >= 71 && dados <= 100) {  
    casilla = casilla + pequeño_salto;  
    if (casilla > 70) {  
        casilla = 70;  
    }  
}  
System.out.println("Posicion L: " + casilla);  
  
/*  
 * Aqui lo que hago es que cuando llega la casilla 70 es que la  
liebre a llegado a la meta  
 * entonces significa que a ganado. Paro el thread de la  
tortuga para que no siga corriendo y compruebo si ella tambien a llegado a la  
misma vez a la meta.  
 * Si a llegado imprimo un empate si no a llegado imprimo que  
la liebre a ganado  
 */  
if (casilla == 70) {  
    Salida.t.stop();  
  
    try {  
        // Me aseguro que el otro thread se a parado para  
determinar el final de la carrera con el join  
        Salida.t.join();  
    } catch (InterruptedException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
    if (Tortuga.casilla == 70) {  
        System.out.println("\n EMPATE");  
    } else {  
        System.out.println("\n La Liebre a llegado a la  
meta y ha ganado!");  
    }  
    break;  
}  
maxCasilla++;  
}  
}  
}
```

// EJECUCIONES



Generalitat de Catalunya
Departament d'Educació
Institut d'Educació Secundària i Superior
d'Ensenyaments Professionals
Esteve Terradas i Illa
Departament d'Informàtica

```
UP2 - Practica_Threads_7\src\threads\Liebre.java - Eclipse IDE
File Edit Refactor Source Navigate Search Project Run Window Help

Practice_Threads_7 src threads Liebre.java

Console
C:\Program Files\Java\jre1.8.0_241\bin\java.exe [13 abr. 2020 18:46:07]
> Position T: 39
> Position T: 40
> Position T: 43
> Position T: 44
> Position T: 47
> Position T: 50
> Position T: 54
> Position T: 47
> Position T: 58
> Position T: 53
> Position T: 56
> Position T: 57
> Position T: 58
> Position T: 62
> Position T: 65
> Position T: 66
> Position T: 62
> Position T: 65
> Position T: 68
> Position T: 69
> Position T: 70
La Tortuga a llegado a la meta y ha ganado!

28
29
30 // DUEÑE
31 } else if (dados >= 11 && dados <= 30) {
32     casilla = casilla + duenne;
33     if (casilla > 70) {
34         casilla = 70;
35     }
36 }
37 // GRAN SALTO
38 } else if (dados >= 31 && dados <= 50) {
39     casilla = casilla + gran_salto;
40     if (casilla > 70) {
41         casilla = 70;
42     }
43 }
44 // RESBALON PEQUEÑO
45 } else if (dados >= 51 && dados <= 70) {
46     casilla = casilla + resbalon_pequeno;
47     if (casilla < 1) {
48         casilla = 1;
49     }
50 }
51 // PEQUEÑO SALTO
52 } else if (dados >= 71 && dados <= 100) {
53     casilla = casilla + pequeno_salto;
54     if (casilla > 70) {
55         casilla = 70;
56     }
57 }
58 System.out.println("Position L: " + casilla);
59
60
```

```
UP2 - Practica_Threads_7\src\threads\Liebre.java - Eclipse IDE
File Edit Refactor Source Navigate Search Project Run Window Help

Practice_Threads_7 src threads Liebre.java

Console
C:\Program Files\Java\jre1.8.0_241\bin\java.exe [13 abr. 2020 18:44:31]
> Position L: 44
> Position L: 53
> Position L: 41
> Position L: 42
> Position L: 43
> Position L: 52
> Position L: 53
> Position L: 51
> Position L: 49
> Position L: 50
> Position L: 51
> Position L: 49
> Position L: 56
> Position L: 59
> Position L: 60
> Position L: 60
> Position L: 61
> Position L: 70
> Position L: 61
> Position L: 62
> Position T: 63
La Liebre a llegado a la meta y ha ganado!

28
29
30 // DUEÑE
31 } else if (dados >= 11 && dados <= 30) {
32     casilla = casilla + duenne;
33     if (casilla > 70) {
34         casilla = 70;
35     }
36 }
37 // GRAN SALTO
38 } else if (dados >= 31 && dados <= 50) {
39     casilla = casilla + gran_salto;
40     if (casilla > 70) {
41         casilla = 70;
42     }
43 }
44 // RESBALON PEQUEÑO
45 } else if (dados >= 51 && dados <= 70) {
46     casilla = casilla + resbalon_pequeno;
47     if (casilla < 1) {
48         casilla = 1;
49     }
50 }
51 // PEQUEÑO SALTO
52 } else if (dados >= 71 && dados <= 100) {
53     casilla = casilla + pequeno_salto;
54     if (casilla > 70) {
55         casilla = 70;
56     }
57 }
58 System.out.println("Position L: " + casilla);
59
60
```



Generalitat de Catalunya
Departament d'Educació
Institut d'Educació Secundària i Superior
d'Ensenyaments Professionals
Esteve Terradas i Illa
Departament d'Informàtica

```
UF2 - Práctica_Threads_7src\threads\Liebre.java - Eclipse IDE
File Edit Refactor Source Navigate Search Project Run Window Help

Practice_Threads_7 src > threads > Liebre.java

Herramienta: Salida (Java Application) C:\Program Files\Java\jre1.8.0_241\bin\java.exe (13 abr. 2020 18:45:24)

> Posición L: 18
> Posición T: 58
> Posición L: 8
> Posición T: 51
> Posición L: 1
> Posición T: 54
> Posición L: 1
> Posición T: 57
> Posición L: 1
> Posición T: 68
> Posición L: 1
> Posición T: 68
> Posición L: 18
> Posición T: 63
> Posición L: 1
> Posición T: 64
> Posición L: 2
> Posición T: 67
> Posición L: 2
> Posición T: 68
> Posición L: 3
> Posición T: 78
> Posición L: 12
> Posición L: 18
La Tortuga a llegado a la meta y ha ganado!

24         casilla = 70;
25     }
26     System.out.println("Posición L: " + casilla);
27
28     /*
29     * Aquí lo que hago es que cuando llega la casilla 70 es que la liebre = llegado = la meta
30     * entonces significa que = ganado. Pero el thread de la tortuga para que no siga corriendo y comiencen a salir también = llegado = la misma vez = la meta.
31     * Si = llegado imprimo un mensaje si no = llegado imprimo que la liebre = ganado
32     */
33     if (casilla == 70) {
34         salida.t.sigue();
35
36         try {
37             // Me aseguro que el otro thread se = parado para determinar el final de la carrera con el join
38             salida.t.join();
39         } catch (InterruptedException e) {
40             // TODO Auto-generated catch block
41             e.printStackTrace();
42         }
43         if (tortuga.casilla == 70) {
44             System.out.println("La TORTUGA");
45         } else {
46             System.out.println("La Liebre a llegado a la meta y ha ganado!");
47         }
48         break;
49     }
50     maxCasilla++;
51 }
52 }
53 }
54 }
55 }
```

```
UF2 - Práctica_Threads_7src\threads\Liebre.java - Eclipse IDE
File Edit Refactor Source Navigate Search Project Run Window Help

Practice_Threads_7 src > threads > Liebre.java

Herramienta: Salida (Java Application) C:\Program Files\Java\jre1.8.0_241\bin\java.exe (13 abr. 2020 18:45:41)

> Posición T: 68
> Posición T: 61
> Posición T: 64
> Posición T: 58
> Posición T: 52
> Posición T: 55
> Posición T: 56
> Posición T: 58
> Posición T: 53
> Posición T: 56
> Posición T: 58
> Posición T: 58
> Posición T: 63
> Posición T: 68
> Posición L: 4
> Posición T: 69
> Posición L: 5
> Posición T: 78
> Posición L: 6
> Posición L: 15
La Tortuga a llegado a la meta y ha ganado!

24         casilla = 70;
25     }
26     System.out.println("Posición L: " + casilla);
27
28     /*
29     * Aquí lo que hago es que cuando llega la casilla 70 es que la liebre = llegado = la meta
30     * entonces significa que = ganado. Pero el thread de la tortuga para que no siga corriendo y comiencen a salir también = llegado = la misma vez = la meta.
31     * Si = llegado imprimo un mensaje si no = llegado imprimo que la liebre = ganado
32     */
33     if (casilla == 70) {
34         salida.t.sigue();
35
36         try {
37             // Me aseguro que el otro thread se = parado para determinar el final de la carrera con el join
38             salida.t.join();
39         } catch (InterruptedException e) {
40             // TODO Auto-generated catch block
41             e.printStackTrace();
42         }
43         if (tortuga.casilla == 70) {
44             System.out.println("La TORTUGA");
45         } else {
46             System.out.println("La Liebre a llegado a la meta y ha ganado!");
47         }
48         break;
49     }
50     }
51     maxCasilla++;
52 }
53 }
54 }
55 }
```



Generalitat de Catalunya
Departament d'Educació
Institut d'Educació Secundària i Superior
d'Ensenyaments Professionals
Esteve Terradas i Illa
Departament d'Informàtica

```
WF2 - Practica_Threads_7\src\threads\Salida.java - Eclipse IDE
File Edit Refactor Source Navigate Search Project Run Window Help
Practica_Threads_7 src threads Jy Salida.java
Console [1]
C:\Program Files\Java\jdk1.8.0_241\bin\java.exe (J3 abr. 2020 18:54:38)
Position L: 55
Position L: 52
Position L: 70
Position T: 42
Position T: 43
Position T: 46
Position T: 49
Position T: 50
Position T: 53
Position T: 44
Position T: 48
Position T: 51
Position T: 54
Position T: 55
Position T: 58
Position T: 59
Position T: 62
Position T: 63
Position T: 64
Position T: 67
Position T: 70
EMULATE
EMULATE
```

```
WF2 - Practica_Threads_7\src\threads\Salida.java - Eclipse IDE
File Edit Refactor Source Navigate Search Project Run Window Help
Practica_Threads_7 src threads Jy Salida.java
Console [1]
C:\Program Files\Java\jdk1.8.0_241\bin\java.exe (J3 abr. 2020 18:56:47)
Position T: 62
Position T: 65
Position T: 68
Position L: 35
Position T: 70
Position L: 44
Position L: 44
Position L: 55
Position L: 41
Position L: 29
Position L: 36
Position L: 39
Position L: 46
Position L: 49
Position L: 50
Position L: 51
Position L: 52
Position L: 61
Position L: 62
Position L: 63
Position L: 63
Position L: 70
EMULATE
EMULATE
```