

Fundamentos de la programación

Grado en Desarrollo de Videojuegos

Examen parcial 1

Indicaciones generales:

- **Lee atentamente** el enunciado e implementa el programa tal como se pide, con los métodos y requisitos que se especifican.
- El programa, además de ser correcto, debe estar bien comentado. Se valorarán la claridad, la concisión y la eficiencia.
- Se entregará un único archivo Program.cs (generado con MonoDevelop) con el programa completo.

En este ejercicio simularemos el comportamiento de un marinero borracho cruzando la pasarela de acceso a su barco. El marinero, avanza paso a paso por la misma en dirección al barco, pero en su estado se tambalea y puede hacer desplazamientos erráticos a izquierda y derecha. Así pues, puede llegar al final de la pasarela y alcanzar el barco, o bien caer por uno de los laterales de la pasarela antes de llegar al barco.

El **ancho** y el **largo** de la pasarela se solicitarán al usuario por teclado y el programa comenzará la simulación sin más intervención del usuario. El marinero puede avanzar recto, a la izquierda o la derecha con igual probabilidad, y la simulación termina cuando el marinero llega al final de la pasarela, o bien se cae al agua y se ahoga. Un ejemplo de ejecución puede ser el siguiente (los caracteres "-" representan la pasarela y "0" el marinero sobre ella):

```
Ancho de la pasarela: 7
Largo de la pasarela: 15
---0---
--0----
---0---
---0---
--0----
-0-----
--0----
-0-----
0-----
0-----
Choff!!!
```

El estado de la simulación viene determinado por las siguientes variables (**ancho**, **largo**, **pasos**, **posicion**, **choff**) (las cuatro primeras de tipo `int` y la última `bool`). En un estado concreto de la simulación contienen la siguiente información:

- **ancho**: ancho de la pasarela. Se solicitará al principio y no cambia durante la simulación.
- **largo**: largo de la pasarela. También se fija al principio y luego no cambia.
- **pasos**: número de pasos que ha dado el marinero. Inicialmente vale 0 y se irá incrementando en cada paso de simulación.
- **posicion**: posición del marinero en la pasarela respecto al extremo izquierdo (1 es el borde izquierdo y **ancho** el borde derecho). Inicialmente el marinero estará en el centro de la pasarela y en cada paso irá variando aleatoriamente: puede incrementarse en 1, decrementarse en 1 o no cambiar, con la misma probabilidad.

- **choff**: variable booleana que indica si el marinero está dentro de la pasarela (**choff=false**) o se ha caído al agua (**choff=true**). (Esta variable es en realidad redundante, pero clarifica la situación).

En el ejemplo anterior, tras tres pasos des simulación tenemos en pantalla:

```
---0---
---0---
--0---
-0---
```

En este estado tenemos: **ancho=7**; **largo=15** (aunque no se refleja en pantalla); **pasos=3** (la primera línea es el estado inicial, **paso=0**); **posicion=2** y **choff=false**.

Se pide implementar los siguientes métodos:

- **static int pideEntero(string s, int min, int max)**: escribe en pantalla la cadena de texto **s** y lee un entero de teclado que debe estar en el intervalo **[min,max]** (tal como se ha hecho en clase);
- **static void dibuja(int ancho, int posicion, bool choff)**: dibuja en pantalla un línea de texto correspondiente al estado actual de la simulación. Si el marinero está situado fuera de la pasarela escribirá "Choff!!!"; en otro caso escribirá la pasarela con caracteres "-" y el marinero con "0" tal como se ha explicado arriba.
- el programa tendrá la siguiente estructura:

```
using System;
namespace marineroborracho {
    class MainClass {
        public static void Main () {
            Random rnd = new Random(); // para aleatorizar la direccion del marinero
            int ancho, largo, posicion, pasos; // estado de la simulacion
            bool choff = false;
            ... // otras variables

            ancho = pideEntero ("Ancho de la pasarela: ",1,10);
            largo = pideEntero ("Largo de la pasarela: ",1,300);
            ...
            dibuja ... // dibujo de la situacion
            ...
            // BUCLE PRINCIPAL DE LA SIMULACION
            while (...) // termina cuando el marinero llega al final o se cae al agua
            {
                // nueva posicion del marinero
                ... rnd.Next (...); // generamos numero aleatorio
                ...
                dibuja ... // dibujo de la situacion
            } // fin while
            // si ha llegado al final, escribir "Has llegado!"
        } // fin Main

        public static int pideEntero(string msg, int min, int max) {
            ...
        } // fin pideEntero

        public static void dibuja(int ancho, int posicion, bool choff) {
            ...
        } // fin dibuja
    }
}
```