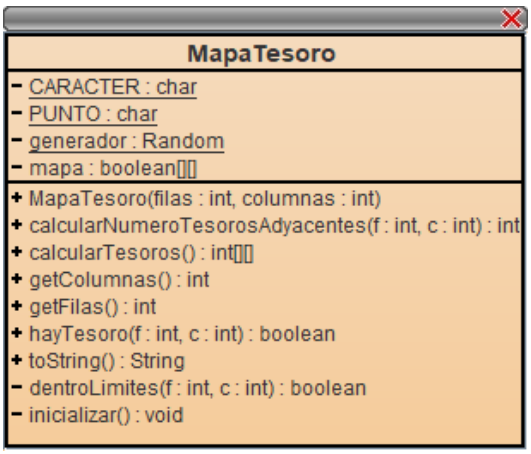
Ejercicio UT5 "*Mapa del tesoro*" (2021-2022) – 2ª evaluación

Ejercicio.

Un mapa del tesoro puede ser representado por una cuadrícula rectangular. En cada posición del mapa puede haber o no un tesoro.

La clase MapaTesoro describe al mapa del tesoro y lo hace a través de un array bidimensional de valores *boolean* que almacenan *true* o *false* si en una posición *f,c* del mapa hay o no un tesoro.

Completa la clase tal como indica el diagrama. La clase TestMapaTesoro incluye código para testearla (se da completa).



Define el atributo *mapa*.

Completa los métodos de la clase MapaTesoro:

a) el constructor - crea el mapa al nº de filas y columnas indicado y lo inicializa con valores *true* / *false* que se asignarán de forma aleatoria con ayuda del método inicializar()

b) inicializar() – inicializa el array mapa a valores aleatorios *true* / *false*

c) getFilas() – devuelve el nº de filas del mapa

d) getColumnas() – devuelve el nº de columnas del mapa

e) toString() - representación textual del mapa de siguiente forma:

mapa

true	false	true	true	false	true
false	false	true	true	true	true
false	true	false	false	true	true
true	true	true	false	false	true

```
  0 1 2 3 4 5
0 $ . $ $ . $
1 . . $ $ $ $
2 . $ . . $ $
3 $ $ $ . . $
```

Cadena devuelta por toString()

Carácter $ donde hay tesoro,
carácter . donde no lo hay

Se incluyen las posiciones de fila y columna

Cada valor formateado a 2 posiciones

f) hayTesoro() – dada una posición *f,c* se devuelve *true* si en la posición indicada hay un tesoro, *false* si la posición no está dentro de los límites del mapa o no hay un tesoro en ella

g) dentroLimites() - método de ayuda que detecta si una posición *f,c* está dentro de los límites del mapa

h) calcularNumeroTesorosAdyacentes() - devuelve el nº de tesoros adyacentes a una posición dada *f,c*. Las posiciones adyacentes en el mapa a *f,c* son las ocho celdas (posiciones) que la bordean (horizontal, vertical y diagonalmente). Un tesoro en la posición dada *f,c* no cuenta como adyacente. El nº de tesoros adyacentes en una posición fuera de los límites es 0.

	0	1	2	3	4	5
0	$		$	$		$
1			$	$	$	$
2		$			$	$
3	$	$	$			$

calcularNumeroTesorosAdyacentes(0, 2) => 3
calcularNumeroTesorosAdyacentes(0, -1) => 0
calcularNumeroTesorosAdyacentes(2, 3) => 5
calcularNumeroTesorosAdyacentes(2, 2) => 5
calcularNumeroTesorosAdyacentes(4, 9) => 0

i) calcularTesoros() - crea y devuelve un array bidimensional de enteros. En este nuevo array se asignará un 9 en la posición *f,c* del mapa en la que haya un tesoro, si no lo hay se asignará el nº de tesoros adyacentes a dicha posición

9	3	9	9	5	9
2	4	9	9	9	9
3	9	5	5	9	9
9	9	9	2	3	9

Posible ejecución

```
Teclee filas del mapa del tesoro:
4
Teclee columnas del mapa del tesoro:
6

Mapa del tesoro:(4 filas , 6 columnas)

   0 1 2 3 4 5
 0 $ . $ $ . $
 1 . . $ $ $ $
 2 . $ . . $ $
 3 $ $ $ . . $

-------------------------------------------------

Hay tesoro en las siguientes posiciones del mapa?
(0, 2): true
(0, -1): false
(2, 3): false
(2, 2): false
(4, 9): false
-------------------------------------------------
```

```
Tesoros adyacentes a las siguientes posiciones del mapa?
(0, 2): 3
(0, -1): 0
(2, 3): 5
(2, 2): 5
(4, 9): 0
-------------------------------------------------

Mapa de enteros
Si hay tesoro en esa posición => 9
Si no hay tesoro en esa posición =>nº de tesoros adyacentes a la misma

 9 3 9 9 5 9
 2 4 9 9 9 9
 3 9 5 5 9 9
 9 9 9 2 3 9
```