SANDRA CIUDAD MORENO 1ºB
Sandra.Ciudad1@alu.uclm.es

ÍNDICE:

1.-ENUNCIADO DEL PROBLEMA ELEGIDO.

7. La reina del ajedrez

En el ajedrez, una reina "come" (o cubre) en horizontal, en vertical y en diagonal.

Por ejemplo, en el tablero de la derecha, la reina de la posición `R', puede "comer" (o cubre) en todas las casillas marcadas con 'x'

x			x			x	
	x		x		x		
		x	x	x			
x	x	x	R	x	x	x	x
		x	x	x			
	x		x		x		
x			x			x	
			x				x

Se pide realizar un programa que, dado un tablero de NxN, donde N es un valor entero positivo que se introduce por teclado, el usuario vaya introduciendo reinas en las filas y columnas que desee hasta que todo el tablero quede cubierto por el poder de las reinas. Para ello, el programa le pedirá al usuario la fila y la columna de la casilla en la que desee colocar una reina. Cada vez que se introduzca una reina en una posición del tablero válida, no cubierta por otra reina introducida previamente, la posición de la reina se marcará con el valor 'R' y todas las casillas que queden cubiertas por dicha reina se marcarán con el valor 'x', mostrando, a continuación, el estado del tablero. Si al introducir una reina en una casilla, ésta ya estuviera cubierta por otra reina introducida antes (es decir, la casilla tiene el valor 'x' o 'R') o cayera fuera de los bordes del tablero, el programa mostrará un mensaje al usuario indicando la situación y no hará nada. El programa finaliza cuando todo el tablero está cubierto por el efecto de las reinas colocadas, mostrando el número de reinas utilizadas.

Siguiendo el ejemplo del tablero anterior, si se introduce una reina en la posición (1,7) el tablero quedaría como en la figura de la derecha

x			x			x	x
x	x	x	x	x	x	x	R
		x	x	x		x	x
x	x	x	R	x	x	x	x
		x	x	x			x
	x		x		x		x
x		x	x			x	x
	x		x				x

Si después se intenta poner una reina en la casilla (3,1), como esta posición ya está "cubierta" (hay una 'x'), el programa no haría nada y mostraría un mensaje indicando que "la casilla (3,1) ya está cubierta". Igualmente, si se intenta colocar una reina en la casilla (8,9), el mensaje indicaría que "la casilla (8,9) está fuera de rango". En ambos casos, el número de reinas utilizadas no se incrementa.

x			x			x	x
x	x	x	x	x	x	x	R
		x	x	x		x	x
x	x	x	R	x	x	x	x
		x	x	x			x
	x		x		x		x
x		x	x			x	x
	x		x				x

Si se colocan sucesivamente varias reinas, por ejemplo, en (0,5), (7,4), (4,0) y (2,1), el tablero quedaría cubierto completamente y el programa terminaría. En este caso, se habrían utilizado 6 reinas.

x	x	x	x	x	R	x	x
x	x	x	x	x	x	x	R
x	R	x	x	x	x	x	x
x	x	x	R	x	x	x	x
R	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	R	x	x	x

2.-ANÁLISIS DEL PROBLEMA.

Para realizar el análisis del problema, hice una lista, en la que especifiqué todos los requisitos pedidos en el enunciado de dicho problema.

REQUISITOS DEL PROBLEMA:

-La reina cubre en horizontal, vertical y diagonal.

-El tablero será de NxN (N es un valor entero positivo introducido por teclado).

-El usuario introduce el número de fila y columna donde quiere poner a la reina:

> -Si la posición elegida es válida, esa casilla se marcará con una R, y las casillas que cubra la reina con x → Se cuentan las reinas y se muestra el tablero.
>
> -Si la posición elegida es no válida puede ser por dos motivos:
>
> > -La casilla ya está ocupada (bien por una R, bien por una x) → Se informa al usuario de que la casilla ya está ocupada.
> >
> > -La casilla está fuera de los límites del tablero → Se informa al usuario de que la casilla está fuera de los límites tablero.

-El programa finaliza cuando todo el tablero esté cubierto por el efecto de las reinas.

-Se muestra las reinas utilizadas para cubrir el tablero

3.-JUSTIFICACIÓN Y REPRESENTACIÓN DE LOS DATOS.

Para la justificación y representación de los datos voy a separar el programa en distintas partes para usar unas capturas de lo que aparece en la consola al ejecutarlo:

En primer lugar, nada más ejecutar el programa y tras introducir las dimensiones de la matriz (que en mi caso he decidido que sea de 8x8) nos aparecerá la matriz vacía. Esto ocurre porque al introducir las dimensiones, se crea la matriz, se recorre rellenándose cada casilla con 'o' y se muestra por pantalla.

A continuación, aparecerán los siguientes mensajes y, tras introducir el número de fila y columna en el que queremos colocar a la reina (que en mi caso he seleccionado la fila 3 y columna 6), se nos mostrará la matriz con una R en la casilla seleccionada para la reina y x en los lugares que esta cubre. Esto se debe a que la casilla seleccionada para la reina es correcta, es decir, que esa casilla no estaba ocupada y se

```
Introduzca la dimensión de la matriz:
8
Se ha creado la siguiente matriz:

 o o o o o o o o
 o o o o o o o o
 o o o o o o o o
 o o o o o o o o
 o o o o o o o o
 o o o o o o o o
 o o o o o o o o
 o o o o o o o o

Introduzca la FILA:
3
Introduzca la COLUMNA:
6

Dato correcto
Se han introducido 1 reinas.

 o o o x o o x o
 o o o o x o x o
 o o o o o x x x
 x x x x x R x x
 o o o o o x x x
 o o o o x o x o
 o o o x o o x o
 o o x o o o x o
```

encuentra dentro del tablero. Además, se mostrará un mensaje que nos indica las reinas que hemos añadido al tablero ya que, al confirmar que los datos anteriores son correctos, se incrementa un contador para las reinas del tablero.

En caso de introducir una casilla ya ocupada, aparecerá lo siguiente. Esto se debe a que se ha recorrido la matriz para colocar a la reina en la casilla seleccionada, y el programa se ha encontrado con que estaba ocupada (bien por la reina o bien por su efecto), por lo que este dato se ha almacenado como no correcto, de manera que ni se ha incrementado el contador de reinas ni ha variado la matriz con respecto a la anterior.

```
El lugar seleccionado ya está ocupado.

Se han introducido 1 reinas.

 o o o x o o x o
 o o o o x o x o
 o o o o o x x x
 x x x x x x R x
 o o o o o x x x
 o o o o x o x o
 o o o x o o x o
 o o x o o o x o
```

En caso de introducir una casilla que no se encuentre en el tablero ocurrirá lo siguiente. En este caso sucede algo muy similar a lo anterior, y es que se recorre la matriz en busca de esa casilla y el programa se da cuenta de que dicha casilla no pertenece al tablero, ya que sobrepasa los límites del mismo, por lo que este dato no se contará como correcto, no incrementando el contador de reinas y manteniendo la matriz en el estado anterior.

```
La casilla seleccionada se encuentra
fuera de los límites del tablero.

Se han introducido 1 reinas.

 o o o x o o x o
 o o o o x o x o
 o o o o o x x x
 x x x x x x R x
 o o o o o x x x
 o o o o x o x o
 o o o x o o x o
 o o x o o o x o
```

Tras introducir todas las reinas necesarias como para completar el tablero con ellas y su efecto, aparecerá el siguiente mensaje y el programa finalizará. Esto se debe a que, una vez en el tablero no quede ningún hueco sin rellenar, la matriz pasará a estar completa, se hará un recuento de las reinas que hay y se dará fin al programa.

```
Se han introducido 5 reinas.

 R x x x x x x x
 x x x x x x x x
 x x x R x x x x
 x x x x x x R x
 x x x x x x x x
 x x x x x x x x
 x x R x x x x x
 x x x x x R x x

La matriz ya está completa.
```

4.-DESCOMPOSICIÓN MODULAR REALIZADA.

A la hora de realizar la descomposición modular, tuve en cuenta cuales iban a ser los procesos que se repetirían en más de una ocasión en el código y aquellos que, aunque no se repitieran, no necesitaban estar en el main.

A continuación, voy a desglosar el código del programa para poder explicar así los métodos realizados y qué funciones tienen.

En primer lugar, se deberán introducir los datos necesarios para crear la matriz vacía donde, posteriormente, se colocarán las reinas y las posiciones cubiertas por ellas, por lo tanto, se mostrará un mensaje indicando al usuario que debe introducir las dimensiones de dicha matriz. Automáticamente se creará una matriz cuadrada con el valor introducido

y se ejecutará el método inicializar, el cual recorre todas las filas y columnas de la matriz para colocar ‘o’ en todas sus casillas, indicando así que estas se encuentran vacías.

Una vez creada la matriz vacía, se pedirá al usuario que introduzca el número de fila y columna en el cual desea colocar una reina, de manera que se ejecutará a continuación el método leerEntero 2 veces, una para la introducción de la fila, y otra para la de la columna. Este método simplemente coge los valores introducidos por el usuario y los lleva al método colocar para comprobar si los datos son o no correctos. Una vez en el método colocar, se determinará si los datos introducidos anteriormente por el usuario son correctos, para ello se comprobará si estos se encuentran dentro de los límites de la matriz creada y si la casilla que se ha seleccionado no está ocupada. En caso de no cumplirse una de las dos condiciones mencionadas anteriormente, los datos se tomarán como incorrectos y se informará al usuario de que, o bien la casilla seleccionada ya está ocupada, o bien no se encuentra dentro de los límites de la matriz creada, mientras que, si se cumplen las dos condiciones, los datos se tomarán como correctos, se incrementará el contador de reinas y se mostrará la matriz, que en este caso contendrá la casilla seleccionada con una R y las casillas que cubre la reina con una x (esto se consigue a través de la ejecución de los métodos rellenarH, rellenarV, rellenarDDerecha, y rellenarDIzquierza).

Para rellenar la matriz con las reinas y las posiciones cubiertas por ellas, se marcará, en primer lugar, la casilla seleccionada por el usuario con una ‘R’ y, posteriormente debemos recurrir al método rellenarH. Este método se encarga de recorrer las columnas de la matriz rellenando con ‘x’ todas aquellas que se encuentren en la misma fila que la R introducida. Luego pasaremos al método rellenarV, que es similar al anterior sólo que, en lugar de recorrer las columnas recorrerá las filas de la matriz, colocando una x en aquellas que se encuentren en la misma columna de la reina. Los métodos relenarDDerecha y rellenarDIzquierda se van a encargar de rellenar con ‘x’ las diagonales que cubre la reina, para ello el método rellenarDDerecha (que es el que rellena las diagonales de la parte derecha de la reina), recorrerá primero las filas y columnas de la parte derecha superior de la reina, incrementando el número de fila y columna al mismo tiempo para encontrar las casillas de la diagonal superior derecha y marcarlas con una ‘x’ y después recorrerá las filas y columnas de la parte inferior derecha decrementando a la vez el número de fila y columna para encontrar así las casillas de la diagonal inferior derecha y rellenarlas con ‘x’.

El método rellenarDIzquierda, hace casi lo mismo que rellenarDDerecha, la única diferencia es que, este irá marcando con una ‘x’ las casillas que se encuentran tanto en la diagonal superior izquierda de la reina como las que se encuentran en la diagonal inferior izquierda de esta. Para ello se sigue el método explicado anteriormente.

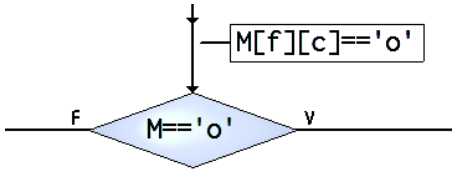
A continuación, se pedirá al usuario que siga introduciendo reinas hasta que en el tablero no quede ninguna casilla vacía. Cuando esto ocurra, el método completa detectará que no hay ninguna casilla en la matriz que contenga ‘o’ y cambiará el estado de la matriz de no completa a completa, dando así por finalizado al programa tras mostrar el número total de reinas empleadas para completar todas las casillas de la matriz.

5.-PSEUDOCÓDIGO.

Para la realización del pseudocódigo del problema que he elegido, he utilizado PSeInt que es una aplicación de programación que permite realizar diagramas de flujo a través bloques que simulan bucles, elementos de entrada, elementos de salida...

El inconveniente que presenta esta aplicación es que está diseñada para estudiantes sin experiencia en programación, por lo que el lenguaje que utiliza es sencillo y muy limitado, de manera que he tenido algunos problemas a la hora de realizar dicho diagrama de flujo, principalmente a la hora de introducir las condiciones en los bucles condicionales (If) y el valor final del contado para los bucles for. Todos estos inconvenientes han sido especificados en el diagrama de flujo a través de comentarios de la siguiente manera:

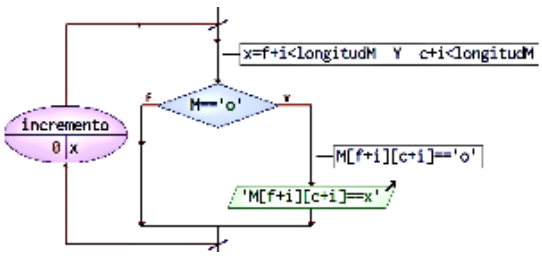
La primera imagen es una representación del condicional de la imagen de abajo, sin embargo, la condición del if en el diagrama de flujo no muestra a M como un array ya que, como he dicho antes, esta aplicación posee un lenguaje muy sencillo y limitado en el que no se encuentran los arrays.
Para solventar este problema he añadido un comentario antes de dicho bucle donde especifica que su condición es realmente M[f][c]==’o’ y no M==’o’



||

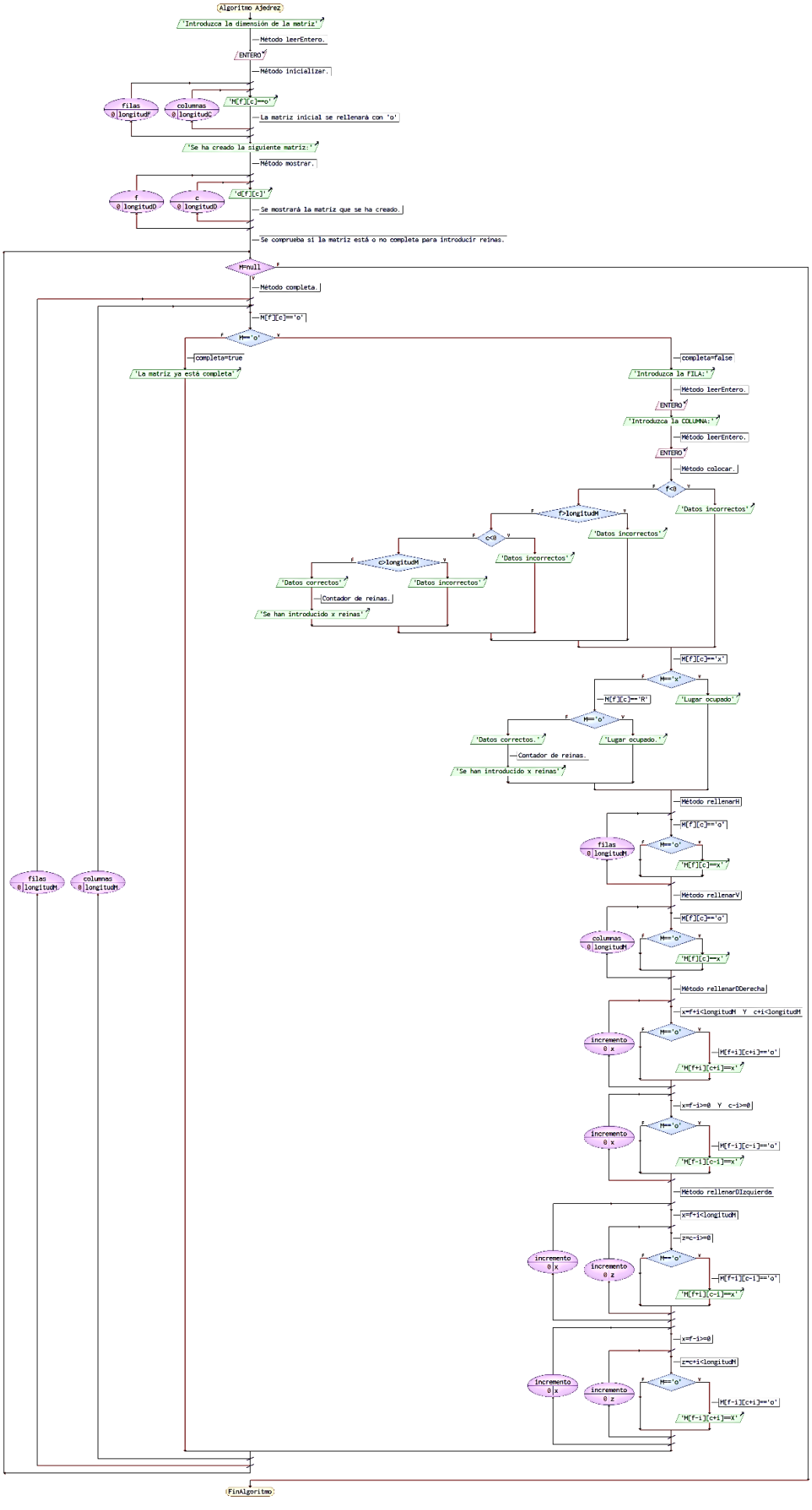
```
if(m[filas][columnas]=='o')
```

En estas imágenes ocurre algo similar a lo anterior.
La primera imagen es la representacion del bucle for de la imagen de abajo, y en este caso el inconveniente sucede cuando hay que seleccionar el valor final para el contador en el bucle for, ya que esta aplicación sólo permite introducir valores constantes. Como solución a este problema, he puesto la constante x como valor final de dicho contador, especificando posteriormente en un comentario el valor de x, ya que, además, este bucle depende de que f-i<longitudM Y de que c+i<longitudM



||

```
for (int i=0; filas +i <M.length && columnas +i<M[0].length; i++)
        if(M[filas+i][columnas+i]=='o')
            M[filas+i][columnas+i]='x';
```


Algoritmo Ajedrez
'Introduzca la dimensión de la matriz'
Método leerEntero.
ENTERO
Método inicializar.
filas
0 longitudF
columnas
0 longitudC
'M[f][c]=='o'
La matriz inicial se rellenará con 'o'
'Se ha creado la siguiente matriz:'
Método mostrar.
f
0 longitudD
c
0 longitudD
'd[f][c]'
Se mostrará la matriz que se ha creado.
Se comprueba si la matriz está o no completa para introducir reinas.
M=null
Método completa.
M[f][c]=='o'
H=='o'
completa=true
'La matriz ya está completa'
completa=false
'Introduzca la FILA:'
Método leerEntero.
ENTERO
'Introduzca la COLUMNA:'
Método leerEntero.
ENTERO
Método colocar.
f<0
'Datos incorrectos'
f>longitudM
'Datos incorrectos'
c<0
'Datos incorrectos'
c>longitudM
'Datos correctos'
'Datos incorrectos'
Contador de reinas.
'Se han introducido x reinas'
M[f][c]=='x'
H=='x'
M[f][c]=='R'
'Lugar ocupado'
H=='o'
'Datos correctos.'
'Lugar ocupado.'
Contador de reinas.
'Se han introducido x reinas'
Método rellenarH
M[f][c]=='o'
filas
0 longitudM
H=='o'
'M[f][c]=='x'
Método rellenarV
M[f][c]=='o'
columnas
0 longitudM
H=='o'
'M[f][c]=='x'
Método rellenarDDerecha
x=f+i<longitudM Y c+i<longitudM
incremento
0 x
H=='o'
M[f+i][c+i]=='o'
'M[f+i][c+i]=='x'
x=f-i>=0 Y c-i>=0
incremento
0 x
H=='o'
M[f-i][c-i]=='o'
'M[f-i][c-i]=='x'
Método rellenarDIzquierda
x=f-i<longitudM
z=c-i>=0
incremento
0 x
incremento
0 z
H=='o'
M[f-i][c-i]=='o'
'M[f-i][c-i]=='x'
x=f-i>=0
z=c+i<longitudM
incremento
0 x
incremento
0 z
H=='o'
M[f-i][c+i]=='o'
'M[f-i][c+i]=='X'
filas
0 longitudM
columnas
0 longitudM
FinAlgoritmo

6.-LECCIONES APRENDIDAS.

Con la realización de este trabajo, en primer lugar, he aprendido a utilizar arrays en Java y a realizar operaciones con ellos, y sobre todo he aprendido a modularizar y a llamar a los métodos en la clase principal, lo que me ha permitido tener el código de mi programa mucho más ordenado y visible, facilitándome así las consultas al mismo. Esto ha hecho que también aprenda a trabajar de forma más ordenada y limpia. De la misma manera he reforzado mis conocimientos sobre bucles (especialmente su estructura y modo de funcionamiento), ya que algunos de ellos no los tenía del todo claros.

Además, también he aprendido a utilizar la aplicación PSeInt para la realización del diagrama de flujo, que me ha hecho ver de una forma más clara qué era lo que realmente se estaba pidiendo en el enunciado del problema, por lo que me ha facilitado la realización del análisis del problema.

Por último, con respecto al ámbito personal, he aprendido que la realización de un programa informático puede ser algo frustrante debido a que, a la hora de realizar el código del mismo, debes tener en cuenta muchas cosas y sobre todo debes tener avanzados conocimientos del tema, por lo que es fácil que se produzcan errores costosos de resolver, pero también he aprendido que con esfuerzo y constancia se puede avanzar progresivamente en este ámbito y que la satisfacción de que cada vez sea más fácil resolver esos “costosos errores” de los que habla antes es muy grande.