SUDOKU (Y TODO LOS DEMÁS)

En esta tarea trabajaremos con **arrays de dos dimensiones, colecciones y flujos de E/S**.

Se trata de hacer **una clase Sudoku**, con una serie de métodos y características que permitirían, mediante el algoritmo asociado, resolverlo. Por el momento nos limitaremos a su declaración y definición de los métodos que podrían resultar de utilidad haciendo uso de colecciones para practicar con ellas.

Sudoku. *Whoami?*

Cortar-pegar. ;-)

Introducción

	4	7		9				
6		8		3		4	9	5
3		5			8			
	3						2	
	1		5	2	7	8	6	
				1	6			
					3		8	
9	5						7	6
				7	4	2		

https://es.wikipedia.org/wiki/Sudoku

Un sudoku es un juego **compuesto por un alfabeto**, normalmente con caracteres numéricos de 1 a 9 (1,… 9) que se **disponen sobre un tablero de NxN, en el que N es el tamaño del alfabeto**, normalmente de 9x9, aunque el alfabeto puede estar formado por letras o ser de otros tamaños.

La disposición de los caracteres del alfabeto debe cumplir las siguientes reglas:

- Los caracteres **no pueden repetirse en cada fila, columna o bloque**.
- Los bloques, por lo tanto, deben tener **N celdas, siendo N el tamaño del alfabeto**.

1	4	7	2	9	5	6	3	8
6	2	8	7	3	1	4	9	5
3	9	5	4	6	8	7	1	2
5	3	6	8	4	9	1	2	7
4	1	9	5	2	7	8	6	3
7	8	2	3	1	6	5	4	9
2	7	1	6	5	3	9	8	4
9	5	4	1	8	2	3	7	6
8	6	3	9	7	4	2	5	1

Por ello, el alfabeto debe ser un cuadrado perfecto, para poder construir bloques de 3x3, 4x4, 5x5,…

Así, los tamaños de los alfabetos y de los sudokus válidos son: 4 (2x2), 9 (3x3), 16 (4x4), 25 (5x5),… Por ejemplo, un sudoku de 16x16:

1	12	15	4	13	3	7	11	14	2	16	9	6	10	8	5
9	16	13	8	14	15	10	4	5	7	6	11	2	12	3	1
7	2	3	10	1	9	6	5	12	15	8	13	14	16	4	11
11	5	14	6	12	2	8	16	3	10	4	1	9	7	15	13
14	9	10	11	6	13	15	1	4	3	12	16	8	5	2	7
2	13	6	3	11	5	16	10	1	8	7	15	4	9	14	12
12	8	1	5	4	7	9	3	6	13	14	2	10	15	11	16
4	15	7	16	2	8	14	12	11	5	9	10	1	3	13	6
16	10	5	14	3	4	11	6	2	1	15	8	7	13	12	9
13	7	4	1	8	12	5	15	10	9	3	6	11	2	16	14
8	6	11	12	9	16	2	13	7	4	5	14	3	1	10	15
15	3	9	2	10	14	1	7	13	16	11	12	5	4	6	8
6	1	12	7	15	10	13	14	9	11	2	4	16	8	5	3
10	11	2	9	16	1	3	8	15	6	13	5	12	14	7	4
3	14	16	15	5	11	4	9	8	12	10	7	13	6	1	2
5	4	8	13	7	6	12	2	16	14	1	3	15	11	9	10

En el ejemplo anterior, el sudoku tiene un alfabeto compuesto por 16 caracteres (1,…16), por lo que es un sudoku de 16x16 con 16 bloques de 4x4.

Lo más usual es representar cada celda con un solo carácter, no número (**un**

carácter, desde un punto de vista de programación, es un número):

7		E		A		3			2		9		0	5	B
4		C	6	E	2		0	D	5		3	A		F	1
F	2	9				5	B	E							D
			3	C	8	7	D	B			A	6			2
8	3	6	1	5		4		2							
		4												3	C
	F		9		A				8		D	1	2		
D		2		1	E	0	9				F				
E	4				7	2	1			6	C	5	F		
9	6		0				5	F		2		C		A	
C	1		7					9	D		E		4		0
				F	9	D			0	4	7				
1		D		2	0	B	F			9				7	5
		0				1					B	D	E		A
2	E	7		9		A	8	0			5	B	C	6	4
B									7			9	1		

El siguiente tamaño válido para el alfabeto sería de 25 caracteres, 5x5:

J	4	N									C		B	2	M	P					E		H		O
H	D		O		6						8			1	A	B	G	C	E		5	L			F
	8		I		A	K	O	3	B	M		L	F	5	1		H	7			C			6	J
B			A				G	L		N	J		H	6	8						D	M	1	2	7
	L	1	5		M		4	2	N			P				D	J		6		9	B	8	A	
F	H		N	O	4	5				D				M	J		I				6		9	C	8
5				M		6	F							K	9	A	C					1		L	
	1				I	2		J	K		7		A	B					N			H	O		
6	A		E	G	9			C		L			O		2	5	7	1	8		F		J	K	M
I	J			K	D	L					1				E	G		3	H					B	5
M	5	3	L	7	N	A	C	I			F	B		G			K	E				O	2	J	H
	F				B	G		O		1	9			E		7		L	5		K	D	6		
K							1			5	O	H			6			9			N				
D	G					J	5	H	3			K	P		B			N			1	C	E	8	
1		C		B	7	F	6	K	D	2		M		N			4		J					5	9
L	I			5			A	E		B		1	7		F		N	J					C		D
8	6	A	H						C	O						I						F	5	7	
3	C	B	1				L		F	9				A	4				7		8	2	N		6
		E	G			7		1	5	C			L			2					H				K
		F			O						H	J		4	C				D		3	E	I	1	L
	N	6	F	H					M	E	K	3				9	P						G	O	2
G	O	5	3	C	P		E	8		F		6							4		B	J	7		I
	9	I	D	8	L	B		6		G			4	H	5	J			C		A		F		1
		J		1	G			F	7				5	9	N	L		2	A			6			C
	B				C			9				A			G		8						K	D	E

Existen sudokus combinados, pero no dejan de ser agrupaciones de sudokus cuadrados. En esta tarea **trabajaremos sudokus de NxN, con N cuadrado perfecto.**

IMPLEMENTACIÓN

La implementación está pensada para realizar sudokus de cualquier tamaño. Para **facilitar la resolución**, sobre todo la comprobación de los bloques del sudoku, **puede hacerse con sudokus de tamaño 9x9**, pero **cumpliendo las especificaciones del enunciado (uso de conjuntos, tipos de datos, comprobaciones, etc.).**

Clase *Sudoku* implementa *Serializable*

La clase Sudoku representa a un Sudoku de cualquier tamaño válido, cuadrado perfecto.

Está formado por un conjunto de ***celdas*** de dos dimensiones de tipo ***char***, además de un **alfabeto**, que es un conjunto (de tipo ***Set***) de elementos de tipo ***Character*** (recordad que las colecciones no puede guardar tipos de datos básicos).

Debe implantar la interface ***Serializable*** para guardar el objeto en un flujo, por medio de una clase de tipo ***ObjectOutputStream***. Dicha interface no tiene métodos que implantar, sólo "informa" a la máquina virtual.

ATRIBUTOS

Contiene dos atributos:

- ***alfabeto***, de tipo ***Set*** de ***Character***.
- ***celas***, un *array* de dos dimensiones de ***char***.

CONSTANTES

- ***ALFABETO_POR_DEFECTO***, conjunto, ***Set***, de alfabeto por defecto: '1', '2', '3', '4', '5', '6', '7', '8', '9'.

CONSTRUCTORES

public ***Sudoku****()*

Construye un Sudoku vacío (cada ***char*** valdrá 0) con el alfabeto por defecto.

public ***Sudoku****(char[] alfabeto)*

Construye un Sudoku vacío (cada ***char*** valdrá 0) el tamaño del alfabeto. Si el alfabeto no es un cuadrado perfecto lanza una excepción (de tipo *Exception*) con el mensaje de "Tamaño de sudoku no válido".

public ***Sudoku****(char[][] celas)*

Construye un Sudoku con las celdas recogidas. Si el *array* de celdas no es cuadrado perfecto de dimensiones posibles lanza una excepción (de tipo *Exception*) con el mensaje de "Tamaño de sudoku no válido".

MÉTODOS

public int ***getSize****()*

Devuelve la anchura del sudoku (número de filas o columnas), calculado a partir de las celdas (no del alfabeto).

public char[][] ***getCeldas****()*

Devuelve el array con las celdas del sudoku.

public void ***setCelda****(int i, int j, char c)*

Asigna el carácter c a la posición i, j del Sudoku. Debe comprobar que no excede los límites del Sudoku.

public void ***setCeldas****(char[][] celdas)*

Asigna las celdas al sudoku. Si el *array* de celdas no es cuadrado perfecto de dimensiones posibles lanza una excepción (de tipo *Exception*) con el mensaje de "Tamaño de sudoku no válido".

public char ***getCelda****(int i, int j)*

Recoge las coordenadas y devuelve el carácter que está en esa posición. Debe comprobar que no excede los límites del Sudoku, en cuyo caso debe devolver 0.

public boolean ***isCompleted****()*

Devuelve si el sudoku tiene todas las celdas ocupadas, sin carácter 0.

public boolean ***isValid****() (25% de la tarea)*

Devuelve verdadero si el Sudoku no tiene valores repetidos en filas, columnas y bloques. Puedes crear lo métodos de ayuda que consideres necesarios para comprobación de filas, columnas y bloques. ***IMPORTANTE: para realizar las comprobaciones de filas, columnas y bloques, debe guardar los elementos en una colección. Si no se hace así se considerará erróneo el método (descuento de 25% de la tarea).***

public java.awt.Point ***getNextVoidCell****()*

Devuelve las coordenadas de la primera celda vacía, aquella con el carácter 0.

public List<Suduku> ***getChildren***()*(25% de la tarea)*

Devuelve un conjunto de posibles Sudokus en la celda vacía. Esto es, un **conjunto de Sudoku que tienen en la primera celda vacía los elementos del alfabeto que no están en la fila, columna o bloque**. *Puede crear métodos intermedios que devuelvan el conjunto de elementos de una fila, columna y bloque, facilitará la escritura del método.*

public boolean ***saveSudoku****(String ruta) (25% de la tarea)*

Guarda el sudoku en un archivo binario recogido como argumento. El objeto debe guardarse por medio de un objeto de tipo *ObjectOutputStream* y su método *writeObject*. *IMPORTANTE: debe crear un* ***FileOutputStream*** *al fichero y encapsularlo en un buffer por medio de la clase* ***BufferredOutputStream****.*

public String ***toString****()*

Devuelve la representación del Sudoku como cadena con el siguiente formato:

513876492
789243651
246915783
328654917
974182536
165397248
697438125
851729364
432561879

Si la celda no tiene valor debe mostrar un espacio.

IMPORTANTE: empléese una clase de tipo StringBuilder para crear la cadena devuelta. No se evaluará correctamente si se hacer de otro modo.

Aplicación *AppSudoku*

Crea una aplicación con un *main* que:

1. **Lea** desde línea de órdenes un **sudoku** con el siguiente formato (cada línea es una fila del sudoku):

   513876492
   789243651
   246915783

328654917
974182536
165397248
697438125
851729364
432561879

2. Indique si es válido, en cuyo caso debe **guardarlo en un archivo** binario, cuyo nombre será pedido desde línea de órdenes.

3. **Muestre los sudokus hijo** (tenga en cuenta que los sudokus no necesariamente deben estar completos, con espacios donde no hay valor). Por ejemplo, otra posible entrada con celdas vacías sería:

5 3876492
789243 5
246 15783
328654917
974182536
1 5397 48
697438125
851729364
4325 1879

ANEXO I. ENLACES DE INTERÉS

https://es.wikipedia.org/wiki/Sudoku
http://www.cs.virginia.edu/~robins/The_Science_Behind_SudoKu.pdf
http://www-m9.ma.tum.de/foswiki/pub/WS2010/CombOptSem/TR_08_04.pdf
https://www.worldpuzzle.org/championships/wsc/ ***(WORLD SUDOKU CHAMPIONSHIP)***
https://www.sudoku-online.org/ ***(EJEMPLOS DE SUDOKUS)***
https://sudoku.com/
http://news.bbc.co.uk/1/hi/world/asia-pacific/6745433.stm
https://arxiv.org/pdf/1802.09660
https://curlie.org/Games/Puzzles/Brain_Teasers/Sudoku