Sudoku : Etape 3

Exercice évolutif d’apprentissage du langage Java et de la conception objet.

# Objectifs

1. Implémentation des contraintes sur les classes.
   * Restrictions sur les valeurs possibles
   * Exceptions si hors limites ou valeur impossible

# Apprentissage

1. Gestion des exceptions.

# Contraintes à implémenter

1. Un sudoku communique l’indice maximum des lignes et des colonnes
2. Un sudoku communique si une position ligne colonne est valide ;
3. Les méthodes setValue() et getValue() soulève une OutOfSudokuException si la position ligne colonne ne correspond pas à une case de Sudoku.
4. Un sudoku communique une valeur est valide.
5. Un sudoku communique la liste des valeurs valides pour les cellules.
6. La méthode setValue() soulève une InvalidSudokuValueException si la valeur en paramètre ne fait pas partie des valeurs valides.

# Descriptif des classes

## SudokuException et sous-classes

1. Classe SudokuException qui étend Exception
   1. Sous-classe InvalidSudokuValueException
   2. Sous-classe OutOfSudokuException

## Sudoku

1. Modification des méthodes setValue() et getValue() pour quelle soulève des exceptions
   1. setValue() OutOfSudokuException et InvalidSudokuValueException.
   2. getValue()OutOfSudokuException.
2. Ajout de méthodes
   1. + getLineSize() : int
   2. + getColumnSize() :int
   3. + isValidPosition(int line, int column) : boolean
   4. + isValidValue(byte value) : boolean
   5. + getValidValues() :byte[] (copie de tabValid pour la protection contre les modifications)

## SudokuV1, SudokuV2, SudokuV3

1. Ajout d’une propriété :
   1. – tabValid : byte []
2. Implémentation des méthodes :
   1. + isValidPosition(int line, int column) : Boolean
   2. + isValidValue (byte value) : boolean
   3. +getValidValues() :byte[]
3. Modification des méthodes :
   1. setValue() :
      1. ajout des throws pour les 2 exceptions.
      2. tester et soulever les exceptions.
   2. getValue() :
      1. ajout du throws de l’exception.
      2. tester et soulever l’exception.

## Main

1. Modification de la méthode : scenario (int type) pour l’affectation des valeurs
   1. Création d’un tableau de byte [][] ou chaque ligne contient 3 valeurs : valeur, ligne, colonne.
   2. Boucle d’affectation via foreach sur chaque ligne.
   3. Pour chaque ligne :
      1. Try catch
      2. Affichage d’un message différent dans les 3 cas :
         1. setvalue ok
         2. position non valide
         3. valeur non valide

# Copie écran





