Sudoku : Etape 5

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

# Objectifs

1. Le sudoku est composé de cellules qui contiennent chacune une valeur

# Apprentissage

1. Enumérés

# Contraintes à implémenter

1. Un sudoku contient un tableau ligne/colonne de cellules
2. Une cellule contient une valeur possible

# Interfaces

1. SudokuValue
2. Cell
   * + getValue() : SudokuValue
   * + setValue(SudokuValue val) *throws InvalidSudokuValueException*
   * +setValue(String val) *throws InvalidSudokuValueException*
     + *cette méthode convertit le String en SudokuValue et appelle* setValue(SudokuValue val)
3. Modifications de l’interface Sudoku
   * Suppression de get/setValue()
   * +getCell(int line, int column) :Cell *throws OutOfSudokuException*
   * +getValidValues() :byte[] devient +getValidValues() :SudokuValue[]
   * +setGrid(Cell[][] grid)

# Classes

## SudokuValue1\_9

Enum implémentant l’interface SudokuValue et représentant les valeurs 1 à 9.

* Propriété : private String display ;
* Override toString()

## SudokuValue1\_4

Enum implémentant l’interface SudokuValue et représentant les valeurs 1 à 4.

* Propriété : private String display ;
* Override toString()

## Cell1\_9

Classe implémentant l’interface Cell.

* Propriété : SudokuValue1\_9 val ;
* + getValue() : SudokuValue
* +setValue(String val) *throws InvalidSudokuValueException*
  + *catch IllegalArgumentException*
* + setValue(SudokuValue val) *throws InvalidSudokuValueException*

## Cell1\_4

Classe implémentant l’interface Cell.

* Propriété : SudokuValue1\_4 val ;
* + getValue() : SudokuValue
* + setValue(String val) *throws InvalidSudokuValueException*
  + *catch IllegalArgumentException*
* + setValue(SudokuValue val) *throws InvalidSudokuValueException*

# Modifications dans les classes existantes

## AbstractSudoku

* Suppression de get/setValue(), changement +getValidValues() :SudokuValue[]
* La propriété byte[] tabValid devient SudokuValue[] tabValid
  + Changement constructeur et Version 1, 2 et 3
* La propriété byte[] grid devient Cell[] grid
  + +setGrid(Cell[][] grid)
  + Changement constructeur et Version 1, 2 et 3
* Implémentation de+getCell(int line, int column) :Cell *throws OutOfSudokuException*

## AbstractSudokuFactory, SudokuFactoryV1, SudokuFactoryV2, SudokuFactoryV3 et SudokuFactoryV4

* Gestion de la création de la grille et appel setGrid de sudoku

## AbstractEcran

Dans afficheEcran on va rechercher la valeur de la cellule de sudoku.