Object Oriented Programming
Exercise Series 6

Exercise 1 We want to implement a small game that takes place on a grid of size $n \times m$ with $2 \leq n, m \leq 10$, where n denotes the number of lines, and m denotes the number of columns. In this game, each cell is either red or blue. Initially every cell starts red. The goal of the game is to color every cell of the grid blue. In order to do that the player can target a cell of the grid to change the color (blue to red, or red to blue) of the cell targeted as well as the cell directly above, below, on its left and on its right, c.f. Figure 1.

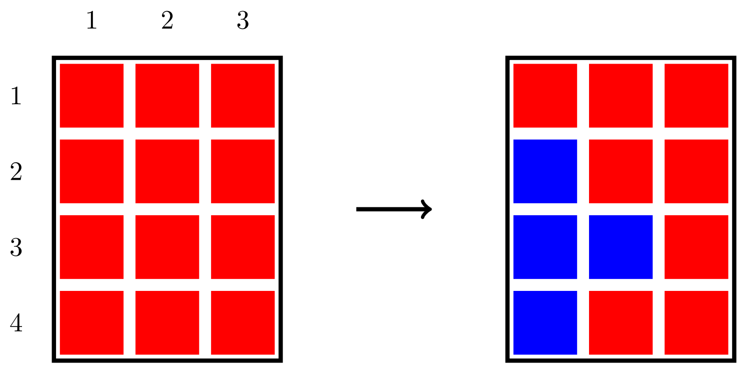

Figure 1: Illustration of targeting the cell $(3, 1)$ (*right*) starting from the initial configuration (left).

Develop a class `ColoringGame` that accurately simulates an instance of this game. It should allow a player to target a given cell in order to change its color as well as its neighbours. It should be able to detect if the game is won.

Test your code with a short sequence of moves. It can be helpful to define a simple display method to obtain a visual representation of the grid at a given time on the Standard Output.

Exercise 2 Now, using the library sliding-puzzle-gui.jar, develop a graphical interface which interacts with `ColoringGame` and allows you to interactively play the game.

You can download this library from the course webpage, in the **Project** section. You can find the library's documentation in the project statement.

Assuming that for some reason you now need to rely on another graphical interface. How does the general architecture of your program impact the implementation of such a modification? In particular, which classes of your program need to change?