Interview Process 02 – Java Assignment

Exercise 1 (Master Mind) Here we propose to create a game of the "Master Mind" type. The aim of the game is to guess a combination of colored pawns chosen by the opponent in as few attempts as possible. The number of pawns (4 or 5) and colors (6 to 8), as well as the possibility of using the same color multiple times and the number of allowed attempts (10 or 12), vary depending on the versions and the level of play. These elements will be customizable here. On each attempt, we give the number of pawns with the correct color but in the wrong position and the number of pawns with the correct color in the correct position. Depending on the level of play, we can either indicate which pawns they are, or only give their respective numbers

. Question 1 (Structure):

Create the different business classes. You can notably create classes for pawns, for a combination of pawns (attempt and secret to be discovered), to represent the result of an attempt (according to the chosen mode), for the board, etc.

Question 2 (Display):

Define a method for displaying the board in a terminal. You can judiciously use Unicode symbols as well as ANSI escape codes for colors.

Question 3 (Validation):

Define a method on the secret to test the validity of an attempt, and return an object representing the result.

Question 4 (Main program):

Write an executable program that allows choosing the number of attempts, pawns, colors, the type of indications, and then playing. The program will display the current state of the board, then read the combination of colors (represented by a number) for the next attempt, as long as the combination is not discovered and there are attempts left. The combination to guess will be defined randomly.

Question 5 (Multi-player mode):

Add the possibility to play with multiple players, taking turns. The system counts the points of each person in the form of the number of remaining attempts. A configurable number of games will be played before ending the game and displaying the total scores.

Question 6 (Save and restore):

Create two symmetrical methods save(Path) and load(Path), allowing to save the current state of the game in a text file on one hand, and to reload a previously saved state on the other hand.

Question 7 :

Implement an automatic player, against whom it will be possible to play

. Question 8 :

Implement a text mode interface (TUI), using for example the Lantern library, or even a graphical interface with elements from the standard Swing library for example.