For Class diagram:

Class Driver will be the top level class, will create a Scanner object, and will ask user “Do you want to play a new game? (Y/N):”, where user will input Y to play and N to not. If yes, Driver will create a Game object with value true if 1 is passed while running, and false if not. Then, runGame with the created Scanner object passed. Will loop through this as many times user wants to play game and input Y.

Class Game will set variables from both provided SecretCodeGenerator and GameConfiugration classes, such as: the answer, how many guesses are allowed, valid colors, pegs. In addition, will create an instance of the History class. In method runGame, while the guessNumber > 0, will read user input and will print “INVALID_GUESS” if input is too long or not valid colors(using Header.validColors()) and if it is, calls static Header method determineMatch to see whether it matches the answer or not. If not a match, calls insertHistory class and decrements guessNumber, else prints “You win!”, and returns.

Class Header has a class called Node, that has variables String guess, int black, white, and boolean match, to describe user input. Has methods validColors to do a O(N^2) check if user input is valid. Method determineMatch returns resulting Node, assigning amount of pegs necessary given user input, and whether if it was a direct match or not.

Class History is constructed with int guesses, in order to instantiate a valid sized array of Nodes. Has void method insertHistory to insert valid guesses into History array, and void method printHistory to print the user input, and the resulting pegs derived from that input.