CSE 241 - OBJECT ORIENTED PROGRAMMING

HOMEWORK 5 CLASS DESIGN DECISIONS REPORT

BoardGame2D (Base Class)

- DATA MEMBERS
  - Vector of characters as "board"
  - Integer for score
  - Pointer for BoardGame2D class
    - This pointer is for determining the which object am I right now. For example, if we are object from PegSolitaire class, then this pointer points to PegSolitaire object and then we can dynamically cast it.
- CONSTRUCTORS
  - No parameter constructor → Assigns score to 0 and pointer to itself
  - 2 parameters constructor → Assigns score and pointer to given values
- FUNCTIONS
  - Pure Virtual Functions
    - playUser takes a string as a parameter and plays the game accordingly.
    - playAuto plays the game by the computer for one move.
    - endGame returns true if the game is ended.
    - initialize initializes the board.
    - print prints the game on the screen starting from the top left corner of the terminal.
  - Overloaded << Operator
    - Prints the game on the screen starting from the top left corner of the terminal.
  - Final Functions
    - playAutoAll plays the game until it is over. Since this is a final function, it cannot be overridden in the derived classes so we need to check the game type first and then play accordingly. It determines the game type by looking at the pointer member of this class. According to its type, it plays the proper game until it ends.
    - Another overload of playUser takes a string inside the function and plays the game accordingly until it is over. This function again checks the game type by looking the object that member pointer is pointing to. This function calls playUser functions of the all derived classes to play 1 move.
  - Static Function
    - playVector function takes a vector of BoardGame2D * objects. It plays all the games in the vector until they end by calling playAutoAll function for all members.
  - Other Helpful Functions
    - boardScore returns an int score value for the current board.
    - increaseScore increases the score by 1.
    - decreaseScore decreases the score by 1.
    - randomGenerator creates a random number between its parameters.

PegSolitaire (Derived Class)

- CONSTRUCTOR
  - No parameter constructor → Assign score to 43 because we have 44 pegs (It is 43 because best score is 0 not 1.), assign pointer to itself, and call initialize function to fill the board.

- FUNCTIONS
  - Inherited Functions
    - playUser takes a string as a parameter and plays the game accordingly. The string that is sent to this function should be in the format "{ROW}{COLUMN}{DIRECTION}" such as "2B UP". Then this function fetchs row, column, and direction from the given string.
    - playAuto plays the game by the computer for one move. Computer randomly chooses 3 numbers: 1 is for row, 1 is for column, and 1 is for direction (1 for up, 2 for down, 3 for left, 4 for right). Then calls one of the helpful functions to move.
    - print prints the game on the screen starting from the top left corner of the terminal.
    - endGame returns true if the game is ended. Function checks all the possible moves that can be done. If there is no possible moves, returns true.
    - initialize initializes the board.

  - Helpful Functions
    - playUp play indicated row and column to the up.
    - playDown play indicated row and column to the down.
    - playLeft play indicated row and column to the left.
    - playRight play indicated row and column to the right.
    - → These play functions also decrease the score by one after each valid move. Because less score means good performance.

EightPuzzle (Derived Class)

- CONSTRUCTOR
  - No parameter constructor → Assign score to 8 (first we assume that initial board is the worst board), assign pointer to itself, and call initialize function to fill the board.

- FUNCTIONS
  - Inherited Functions
    - playUser takes a string as a parameter and plays the game accordingly. Firstly, this function finds the place of the empty slot because it will be moved in the board. The string that is sent to this function should be in the format "{DIRECTION}" such as "UP". Then this function call the proper play function.
    - playAuto plays the game by the computer for one move. Computer randomly chooses 1 number for direction (1 for up, 2 for down, 3 for left, 4 for right). Then calls one of the play functions to move.
    - print prints the game on the screen starting from the top left corner of the terminal.
    - endGame returns true if the game is ended. Function checks if the board is in order. It returns true if it is in order.
    - initialize initializes the board. First initialize it in order, then shuffle it until it becomes solvable.

  - Other Helpful Functions
    - playUp play indicated row and column to the up.
    - playDown play indicated row and column to the down.
    - playLeft play indicated row and column to the left.
    - playRight play indicated row and column to the right.
    - calcScore calculates the score. If all the elements are in their place, then this function assign 0 to score. It increases the score for every incorrect element place.
    - isSolvable checks if the board can be solved. It looks all the elements one by one and find the smaller values from the current element in the remaining elements. Then add those number of smaller elements for each element. If the sum is even then board can be solved, if not it cannot.

    → Play functions also call calcScore function to calculate score after each valid move.

Klotski (Derived Class)

- CONSTRUCTOR
  - No parameter constructor → Assign score to 4 (According to distance of the big rectangle to the goal.), assign pointer to itself, and call initialize function to fill the board.

- FUNCTIONS
  - Inherited Functions
    - playUser takes a string as a parameter and plays the game accordingly. The string that is sent to this function should be in the format "{ROW}{COLUMN}{DIRECTION}" such as "2B UP". Then this function fetchs row, column, and direction from the given string.
    - playAuto plays the game by the computer for one move. Computer randomly chooses 3 numbers: 1 is for row, 1 is for column, and 1 is for direction (1 for up, 2 for down, 3 for left, 4 for right). Then calls one of the helpful functions to move.
    - print prints the game on the screen starting from the top left corner of the terminal.
    - endGame returns true if the game is ended. Function checks the place of big square. If it is in the correct place (which is middle of the bottom row), function returns true.
    - initialize initializes the board.

  - Other Helpful Functions
    - playUp play indicated row and column to the up.
    - playDown play indicated row and column to the down.
    - playLeft play indicated row and column to the left.
    - playRight play indicated row and column to the right.
    - calcScore calculates the score. It calculates the distance of the big square to the correct place. If square is in the correct place, then score becomes 0.

    → Play functions also call calcScore function to calculate score after each valid move.