Assignment 4

Mesk Alhamaideh

Section 1

Chess Game Design

Introduction

Assignment4 code is a design for a chess game between two players, the code does not validate the movements of each player assuming they enter valid moves, and only focuses on the design.

Object Oriented Design

The code consists (ChessGame, ChessBoard, Main, Player) classes. “Piece” is an interface and the following classes (King, Queen, Bishop, Knight, Rook) implement it.

ChessGame class is responsible for keeping tabs on the players turns, validating the string for the move, passing the move to ChessBoard class if it’s valid, checking if the game is done and announcing the winner.

ChessBoard class has an 8*8 matrix of Pieces to keep tabs on the location of each piece, ChessBoard follows the Singleton design pattern because we only need one board in a game of chess.

Piece interface has four methods, one to return the piece type, one to validate a move and has a different implementation depending on the piece type and set and get white which is a boolean that tells us if the piece is black or white.

Player class stores the name of the player and which color he is.

Clean Code

- The code is DRY: "Every piece of knowledge must have a single, unambiguous, authoritative representation within a system".
- Meaningful variable and function names like isBlackCheckMated and movePiece.
- Functions are short and don't need many arguments.
- No long comments
- YANGI (you are not gonna need it): I made sure to only write needed functions and not over engineer.

SOLID Principles

- Single responsibility: each module in the code has only one responsibility
- Dependency inversion: "High-level modules should not depend on low-level modules. Both should depend on abstractions"

  Which is achieved by making ChessBoard have a 2d matrix of Pieces (an interface)

- Open Closed principle: the code is extendable if any changes occur.

Effective Java

- Prefer interfaces to abstract classes: which is why I made Piece an interface instead of an abstract class
- Minimize the accessibility of classes and members: did that by making data members of each class private.