

[previous](#) [up](#) [next](#)

Lab No.7

Chess

Define a class called **Piece**. This will represent a chess piece (a figure on the chessboard). Store its position and its color (Black/White) and its `__repr__` function should write the position like A2, G3 etc. Store the position as integers.

- Define child classes **King** and **Pawn**!
- Implement their `__repr__` method which also prints the type of the figure. (You can implement it in the parent class, too)
 - ♦ Look the unicode characters of the figures: [\[1\]](#)
 - ♦ Or their one letter names
- Every child class should have a `.move(pos)` method, where **pos** is a string like A3, G2 etc. Move the piece to that position if it is accepted by the chess rules. If the move was successful return **True** otherwise **False**!
- Define a **PieceMoveError** exception. If the move is not allowed, then raise this exception!

Implement the **Board** class. Store the list of figures.

- Make an **add** method which puts a figure on the table.
- Implement a `move(pos1, pos2)` method in **Board** which moves a figure from one position to the other, if the move is allowed (handle the **PieceMoveError** exception).
 - ♦ Mind that there may be an other figure on the **pos2** position. If the target position is occupied with the same player's figure, then the move is not allowed. If the target is the other player's figure, then remove their figure and replace it with the new figure (its a *capture*).
 - ♦ Its easier if you implement an `.occupied(pos)` method first, which tells whether a given position is occupied or not.
- Implement the other pieces: **Knight, Rook, Bishop** and **Queen**!
 - ♦ Start with the **Rook**, its move is the simplest. Mind that the figure cannot pass through existing figures!
 - ♦ Implement the **Knight** class! The Knight and Queen can jump through other figures, so its not a problem for them.
 - ♦ After the bishop its not hard to implement **Queen**.
- Define a **check** method in Board. Return the color of the player who's King is in a chess position, or empty string of there is no chess position!
 - ♦ Modify the Board's move method not to allow self-checks (when the active player makes its own King into a chess)!
- Implement the `__repr__` method of **Board**!

After all these, implement the **start** method which assigns the board to a starting position.

CLI

- Make a chess game in command line: read the moves of the players with `raw_input` and print the board after each step. The white player starts and mind that the same player cannot move twice.
 - ♦ It's better to use algebraic notation (Bf5, Qc3, Ne2, Kcd4, Kxd5 etc.)
- Try it from command line

[previous](#) [up](#) [next](#)