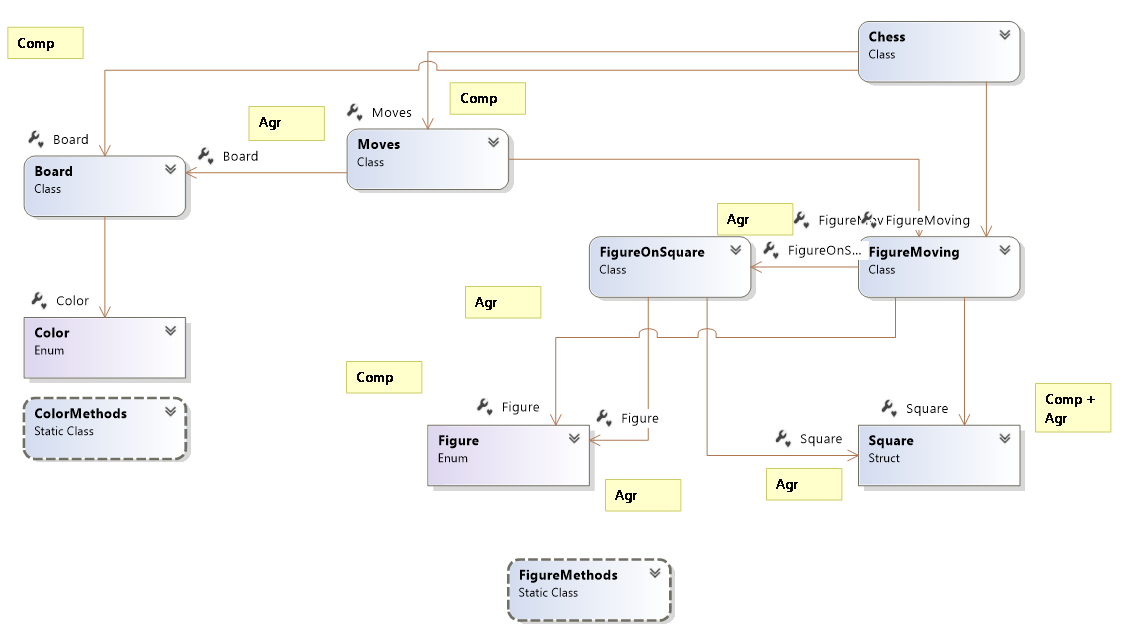
Тема : Шахи

UML Diagram:



Код програми :

Класс Chess : using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ChessLibrary

{

public class Chess

{

static void Main(string[] args)

{

}

public string fen { get; private set; }

Board board;

Moves moves;

public Chess(string fen = "rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1")

{

this.fen = fen;

board = new Board(fen);

moves = new Moves(board);

}

Chess(Board board)

{

this.board = board;

fen = board.fen;

moves = new Moves(board);

}

public Chess Move(string move)

{

FigureMoving fm = new FigureMoving(move);

if (!moves.CanMove(fm))

return this;

Board nextboard = board.Move(fm);

Chess nextChess = new Chess(nextboard);

return nextChess;

}

public char GetFigureAt(int x , int y)

{

Square square = new Square(x, y);

Figure f = board.GetFigureAt(square);

return f == Figure.none ? '.' : (char)f;

}

internal FigureMoving FigureMoving

{

get => default(FigureMoving);

set

{

}

}

internal Moves Moves

{

get => default(Moves);

set

{

}

}

internal Board Board

{

get => default(Board);

set

{

}

}

}

}

Класс Board

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ChessLibrary

{

class Board

{

public string fen { get; set; }

Figure[,] figures;

public Color moveColor { get; private set; }

public int moveNumber { get; private set; }

public Board(string fen)

{

this.fen = fen;

figures = new Figure[8, 8];

Init();

}

void Init()

{

//"rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1"

// 0 1 2 3 4 5

string[] parts = fen.Split();

if (parts.Length != 6) return;

InitFigures(parts[0]);

moveColor = (parts[1] == "b") ? Color.black : Color.white;

moveNumber = int.Parse(parts[5]);

//moveColor = Color.white;

}

//\*\*

void InitFigures (string data)

{

for (int j = 8; j >= 2; j--)

data = data.Replace(j.ToString(), (j - 1).ToString() + "1");

data = data.Replace("1", ".");

string[] lines = data.Split('/');

for (int y = 7; y >= 0; y--)

for (int x = 0; x < 8; x++)

figures[x, y] = lines[7 - y][x] == '.' ? Figure.none :

(Figure)lines[7 - y][x];

}

public Figure GetFigureAt(Square square)

{

if (square.OnBoard())

return figures[square.x, square.y];

return Figure.none;

}

void SetFigureAt(Square square , Figure figure)

{

if (square.OnBoard())

figures[square.x, square.y] = figure;

}

void GenerateFen()

{

fen = FenFigures() + " " +

(moveColor == Color.white ? "w" : "b") +

" - - 0 " + moveNumber.ToString();

}

string FenFigures()

{

StringBuilder sb = new StringBuilder();

for (int y = 7; y >= 0; y--)

{

for (int x = 0; x < 8; x++)

sb.Append(figures[x, y] == Figure.none ? '1' : (char)figures[x, y]);

sb.Append('/');

}

string eight = "11111111";

for (int j = 8; j >= 2; j--)

sb.Replace(eight.Substring(0, j), j.ToString());

return sb.ToString();

}

public Board Move (FigureMoving fm)

{

Board next = new Board(fen);

next.SetFigureAt(fm.from, Figure.none);

next.SetFigureAt(fm.to, fm.promotion == Figure.none ? fm.figure : fm.promotion);

if (moveColor == Color.black)

next.moveNumber++;

next.moveColor = moveColor.FlipColor();

next.GenerateFen();

return next;

}

internal Color Color

{

get => default(Color);

set

{

}

}

}

}

Перечисление Color

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ChessLibrary

{

enum Color

{

none,

white,

black

}

Класс ColorMethods

static class ColorMethods

{

public static Color FlipColor(this Color color)

{

if(color == Color.black) return Color.white;

if (color == Color.white) return Color.black;

return Color.none;

}

}

}

Перечисление Figure

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ChessLibrary

{

enum Figure

{

none,

whiteKing = 'K',

whiteQueen = 'Q',

whiteRook = 'R',

whiteBishop = 'B',

whiteKnight = 'N',

whitePawn = 'P',

blackKing = 'k',

blackQueen = 'q',

blackRook = 'r',

blackBishop = 'b',

blackKnight = 'n',

blackPawn = 'p'

}

Класс FigureMethods

//extensions

static class FigureMethods

{

public static Color GetColor (this Figure figure)

{

if (figure == Figure.none)

return Color.none;

return (figure == Figure.whiteKing ||

figure == Figure.whiteKnight ||

figure == Figure.whiteBishop ||

figure == Figure.whiteQueen ||

figure == Figure.whiteRook ||

figure == Figure.whitePawn)

? Color.white : Color.black;

}

}

}

Класс FigureMoving

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ChessLibrary

{

class FigureMoving

{

public Figure figure { get; private set; }

public Square from { get; private set; }

public Square to { get; private set; }

public Figure promotion { get; private set; }

internal Square Square

{

get => default(Square);

set

{

}

}

internal FigureOnSquare FigureOnSquare

{

get => default(FigureOnSquare);

set

{

}

}

internal Figure Figure

{

get => default(Figure);

set

{

}

}

public FigureMoving(FigureOnSquare fs, Square to, Figure promotion = Figure.none)

{

figure = fs.figure;

from = fs.square;

this.to = to;

this.promotion = promotion;

}

public FigureMoving (string move) //Pe2e4

{

figure = (Figure)move[0];

from = new Square(move.Substring(1, 2));

to = new Square(move.Substring(3, 2));

promotion = (move.Length == 6) ? (Figure)move[5] : Figure.none;

}

}

}

Класс FigurOnSquare

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ChessLibrary

{

class FigureOnSquare

{

public Figure figure { get; private set; }

public Square square { get; private set; }

internal FigureMoving FigureMoving

{

get => default(FigureMoving);

set

{

}

}

internal Figure Figure

{

get => default(Figure);

set

{

}

}

internal Square Square

{

get => default(Square);

set

{

}

}

public FigureOnSquare(Figure figure,Square square)

{

this.square = square;

this.figure = figure;

}

}

}

Класс Moves

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ChessLibrary

{

class Moves

{

FigureMoving fm;

Board board;

public Moves(Board board)

{

this.board = board;

}

internal FigureMoving FigureMoving

{

get => default(FigureMoving);

set

{

}

}

internal Board Board

{

get => default(Board);

set

{

}

}

public bool CanMove(FigureMoving fm)

{

this.fm = fm;

return

CanMoveTo() &&

CanMoveFrom() &&

CanFigureMove();

}

bool CanMoveFrom()

{

return fm.from.OnBoard() &&

fm.figure.GetColor() == board.moveColor;

}

bool CanMoveTo()

{

return fm.to.OnBoard() &&

board.GetFigureAt(fm.to).GetColor() != board.moveColor;

}

bool CanFigureMove()

{

return true;

}

}

}

Класс Square

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ChessLibrary

{

struct Square

{

public static Square none = new Square(-1,-1);

public int x { get; private set; }

public int y { get; private set; }

public Square(string e2)

{

if (e2.Length==2 &&

e2[0] >= 'a' &&

e2[0] <= 'h' &&

e2[1] >= '1' &&

e2[1] <= '8')

{

x = e2[0] - 'a';

y = e2[1] - '1';

}

else

this = none;

}

public Square (int x,int y)

{

this.x = x;

this.y = y;

}

public bool OnBoard()

{

return x >= 0 && x < 8 &&

y >= 0 && y < 8;

}

}

}

Класс Program

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using ChessLibrary;

namespace DemoChess

{

class Program

{

static void Main(string[] args)

{

Chess chess = new Chess();

while (true)

{

Console.WriteLine(chess.fen);

Print(ChessToAscii(chess));

string move = Console.ReadLine();

if (move == "") break;

chess = chess.Move(move);

}

}

static string ChessToAscii(Chess chess)

{

string text = " +-----------------+\n";

for (int y = 7; y >= 0; y--)

{

text += y + 1;

text += " | ";

for (int x = 0; x < 8; x++)

text += chess.GetFigureAt(x, y) + " ";

text += "|\n";

}

text += " +-----------------+\n";

text += " a b c d e f g h\n ";

return text;

}

static void Print(string text)

{

ConsoleColor oldForeColor = Console.ForegroundColor;

foreach (char x in text)

{

if (x >= 'a' && x <= 'z')

Console.ForegroundColor = ConsoleColor.Red;

else if (x >= 'A' && x <= 'Z')

Console.ForegroundColor = ConsoleColor.White;

else

Console.ForegroundColor = ConsoleColor.Cyan;

Console.Write(x);

}

Console.ForegroundColor = oldForeColor;

}

}

}

Висновок : Під час виконання лабораторно роботи я вдосконалив навички програмування . Вивчив UML діаграми (види зв’язків між класами).