

УДУ імені Михайла Драгоманова
Факультет математики, інформатики та фізики

Кафедра комп'ютерної та програмної інженерії

Лабораторна робота №3
з дисципліни
«Архітектура програмної платформи .Net »

Виконав: Стасюк Євгеній М.

Група: 41ПЗ

Факультет математики, інформатики та фізики

Викладач: Січкаренко В.О.

Київ – 2024

```

using System;
using System.Collections.Generic;

namespace Lab3
{
    Ссылка: 15 | 0 изменений | 0 авторов, 0 изменений
    class ChessPiece
    {
        Ссылка: 2 | 0 изменений | 0 авторов, 0 изменений
        public string Name { get; set; }
        Ссылка: 2 | 0 изменений | 0 авторов, 0 изменений
        public string Color { get; set; }
        Ссылка: 2 | 0 изменений | 0 авторов, 0 изменений
        public int b { get; set; }
        Ссылка: 2 | 0 изменений | 0 авторов, 0 изменений
        public int d { get; set; }

        Ссылка: 4 | 0 изменений | 0 авторов, 0 изменений
        public ChessPiece(string name, string color, int b, int d)
        {
            this.Name = name;
            this.Color = color;
            this.b = b;
            this.d = d;
        }
    }

    Ссылка: 3 | 0 изменений | 0 авторов, 0 изменений
    class ChessBoard
    {
        private List<ChessPiece> _pieces;

        Ссылка: 1 | 0 изменений | 0 авторов, 0 изменений
        public ChessBoard()
    }
}

```

```

    }

    Ссылка: 3 | 0 изменений | 0 авторов, 0 изменений
    class ChessBoard
    {
        private List<ChessPiece> _pieces;

        Ссылка: 1 | 0 изменений | 0 авторов, 0 изменений
        public ChessBoard()
        {
            _pieces = new List<ChessPiece>();
        }

        Ссылка: 4 | 0 изменений | 0 авторов, 0 изменений
        public void AddPiece(ChessPiece piece)
        {
            _pieces.Add(piece);
        }

        Ссылка: 1 | 0 изменений | 0 авторов, 0 изменений
        public List<ChessPiece> GetPieces()
        {
            return _pieces;
        }

        Ссылка: 1 | 0 изменений | 0 авторов, 0 изменений
        public bool IsMoveValid(ChessPiece piece, int newX, int newY)
        {
            // TODO: реалізувати перевірку валідності ходу
            return true;
        }
    }

    Ссылка: 0 | 0 изменений | 0 авторов, 0 изменений
    class Program
    {
        Ссылка: 0 | 0 изменений | 0 авторов, 0 изменений
        static void Main(string[] args)
        {
            ChessBoard board = new ChessBoard();

            // Створимо фігури
            ChessPiece whitePawn = new ChessPiece("Пішак", "Білий", 1, 1);
            ChessPiece blackPawn = new ChessPiece("Пішак", "Чорний", 6, 1);
            ChessPiece whiteRook = new ChessPiece("Рок", "Білий", 0, 0);
            ChessPiece blackRook = new ChessPiece("Рок", "Чорний", 7, 0);

            // Додаємо фігури на шаховицю
            board.AddPiece(whitePawn);
            board.AddPiece(blackPawn);
            board.AddPiece(whiteRook);
            board.AddPiece(blackRook);

            // Друкуємо шаховицю
            foreach (ChessPiece piece in board.GetPieces())
            {
                Console.WriteLine("{0} {1} {2} {3}", piece.Name, piece.Color, piece.b, piece.d);
            }

            // Перевіримо валідність ходу
            bool validMove = board.IsMoveValid(whitePawn, 2, 2);
            Console.WriteLine("Валідність ходу: {0}", validMove);
        }
    }
}

```

```

        ChessBoard board = new ChessBoard();

        // Створимо фігури
        ChessPiece whitePawn = new ChessPiece("Пішак", "Білий", 1, 1);
        ChessPiece blackPawn = new ChessPiece("Пішак", "Чорний", 6, 1);
        ChessPiece whiteRook = new ChessPiece("Рок", "Білий", 0, 0);
        ChessPiece blackRook = new ChessPiece("Рок", "Чорний", 7, 0);

        // Додаємо фігури на шаховицю
        board.AddPiece(whitePawn);
        board.AddPiece(blackPawn);
        board.AddPiece(whiteRook);
        board.AddPiece(blackRook);

        // Друкуємо шаховицю
        foreach (ChessPiece piece in board.GetPieces())
        {
            Console.WriteLine("{0} {1} {2} {3}", piece.Name, piece.Color, piece.b, piece.d);
        }

        // Перевіримо валідність ходу
        bool validMove = board.IsMoveValid(whitePawn, 2, 2);
        Console.WriteLine("Валідність ходу: {0}", validMove);
    }
}

```

Цей код реалізує просту модель шахової дошки. Він визначає два класи: ChessPiece для представлення шахової фігури та ChessBoard для представлення шахової дошки.

Клас ChessPiece має чотири властивості:

1. Name - назва фігури
2. Color - колір фігури
3. b - координата x фігури
4. d - координата y фігури

Клас ChessBoard має один приватний список `_pieces`, який містить усі фігури на дошці. Він має три методи:

1. • `AddPiece()` - додає фігуру на дошку
2. • `GetPieces()` - повертає список усіх фігур на дошці
3. • `IsValidMove()` - перевіряє валідність ходу фігури
4. Метод `Main()` створює нову шахову дошку і додає на неї чотири фігури:
5. • Білого пішака на позицію (1, 1)
6. • Чорного пішака на позицію (6, 1)
7. • Білого тура на позицію (0, 0)
8. • Чорні тура на позицію (7, 0)

Далі код друкує стан шахової дошки. На консоль виводиться наступне:

Пішак Білий 1 1

Пішак Чорний 6 1

Рок Білий 0 0

Рок Чорний 7 0

Нарешті, код перевіряє валідність ходу білого пішака на позицію (2, 2). В цьому випадку хід є дійсним, тому метод `IsValidMove()` повертає `true`.