**Документация по рефактуриране на проект “Game-Fifteen-7”**

*Team "Game-Fifteen-7"*

1. Структура на проекта:

- Частта от Main() метода, която отговаря за принтирането на ранглистата е отделена в отделен метод PrintScoreboard().

- Частта от Main() метода, която отговаря за добавяне на нов играч завършил играта към ранглистата е отделена в отделен метод AddNewTopPlayer(inputOfPlayerName).

- Частта от Main() метода, която отговаря за принтиране на текста който казва, че играта е спечелена е отделена в отделен метод PrintTheGameIsWon();

- Частта от Main() метода, която отговаря за изпълнението на въведените команди от играча е отделена в отделен метод ExecuteTheGameCommand(inputCommand).

- Частта от Main() метода, която отговаря за рендериране на игралното поле е отделена в отделен метод PrintTheGameField();

- Частта от Main() метода, която отговаря за принтиране на текста който показва началните надписи и обяснения за играта е отделена в отделен метод PrintWelcomeMessage();

- Частта от Main() метода, която отговаря за рестартиране на играта и зануляване на броя ходове направени до сега в играта и преподреждане на полето е отделена в отделен метод StartNewGame();

- Частта от Main() метода, която отговаря за разбъркване на цифрите по полето преди игра е отделена в отделен метод ShuffleThePuzzleField();

- Статик полетата в началото на кода са преместени в новият клас GameEngine и са направени на „private“.

- Кода, който се намира в Main() метода е преместен в метода StartTheGame() в класа GameEngine.

- Метода StartNewGame() е преместен в класа GameEngine.

- Метода ShuffleThePuzzleField() е преместен в класа GameEngine.

- Метода CheckIsTheMoveAreLegal(int row, int col) е преместен в класа GameEngine.

- Метода MoveTheNumberOfField(int number) е преместен в класа GameEngine.

- Метода ExecuteTheGameCommand(string inputCommand) е преместен в класа GameEngine.

- Метода IsPuzzleSolved() е преместен в класа GameEngine.

- Метода PrintWelcomeMessage() е преместен в класа GameEngine.

- Метода PrintTheGameField(int[,] puzzleField)е преместен в класа ConsolePrinter.

- Метода PrintScoreboard(int countOfTopPlayers, List<Tuple<string, int>> topPlayersScores) е преместен в класа ConsolePrinter.

- Метода PrintTheGameIsWon(int countOfTotalMoves)е преместен в класа ConsolePrinter.

- Променливата „private static readonly Random RandomGenerator“, която се намира в класа „GameEngine“ е преместена в RandomShuffle в метода „Shuffle(PuzzleField puzzleField)“ като „Random randomGenerator“.

- Създаден е обекта Cell, който се ползва като обект в игралното поле.

- Създаден е класа CommandManager, който отговаря за изпълнението на дадена команда.

- Създаден е класа ExitCommand, който отговаря за излизането от играта.

- Метода ShuffleThePuzzleField() е премахнат от класа “GameEngine” и е преместен в класа „ShuffleStrategy“ като метод с името „Shuffle“.

- Създаден е интерфейса “ICell”, който съдържа

int Context { get; set; }

int Row { get; set; }

int Col { get; set; }

- Създаден е интерфейса “ICommand”, който съдържа

void Execute();

- Създаден е обекта „PuzzleField“.

- Създаден е класа „RestartCommand“, който отговаря за рестартиране на играта.

- Създаден е класа „TopCommand“, който отговаря за възпроизвеждане на командата „top“.

- Поради преизползването на някой части код отговарящи за модификациите върху игралното поле са изкарани в нов клас „PuzzleFieldManager“.

- Добавен е класа „PlayerAdapter“ заради имплементацията на “Adapter” дизайн патерна.

- Добавен е класа „PrintablePlayer“ заради имплементацията на “Adapter” дизайн патерна.

- Метода „RearrangePuzzleField“ е преместен от класа „RandomShuffle“ в класа „PuzzleFieldManager“.

2) Сорс код:

- Референциите (using System, using System.Collections.Generic, using System.Linq, using System.Text) са преместени след неимспейса.

- В масива “static int[,] a = new int[4, 4] {{1,2,3,4}, {5,6,7,8}, {9,10,11,12}, {13,14,15,0}};“, са добавени празни пространства между елементите на масива и разделителните запетаи – “static int[,] a = new int[4, 4] { { 1, 2, 3, 4 }, { 5, 6, 7, 8 }, { 9, 10, 11, 12 }, { 13, 14, 15, 0 } };”.

- Премахнати са ненужните празни пространства между променливите.

- Добавени са празни пространства преди и след условните конструкции и циклите.

- Множеството променливи декларирани на един ред са разделени на отделни самостоятелно на отделни редове.

- Добавени са къдрави скоби където липсват преди и след тялото на условните оператори.

- Рандом генератора е изведен като „Private static readonly“ поле в началото на кода.

- Метода “static bool check(int i, int j)” e преименуван на “static bool CheckIsTheMoveAreLegal(int row, int col)”.

- Метода „static void Move(int number)“ е преименуван на „static void MoveTheNumberOfField(int number)“.

- Метода „static bool check2()“ e преименуван на „static bool IsPuzzleSolved()“.

- Метода „static void startagain()“ е преименуван на „StartNewGame()“.

- Кода който се използва за разбъркване на числата в игровото поле   
  
for (int i = 0; i < 1000; i++)  
{  
 ………  
{  
  
и се съдържа в методите „StartNewGame()“ и във първият "while“ цикъл ползван в Main() е отделен като отделен метод ShuffleThePuzzleField().

- Кода, който управлява листа с топ играчите в играта не работи коректно и за това метода „static void move(int i, string res)“ се премахва, защото се ползва от горепосочения код, а самия код който менажира резултатите на топ играчите е пренаписан.

- Кода, който изкарва изходни данни с информация за топ играчите до момента е пренаписан от

(countOfTopPlayers != 0)  
{  
 for (int i = 5 - countOfTopPlayers; i < 5; i++)  
 {  
 Console.WriteLine("{0}", topPlayersScores[i]);  
 }  
 }  
  
на  
  
(countOfTopPlayers != 0)  
{  
 for (int i = 0; i <= countOfTopPlayers - 1; i++)  
 {  
 Console.WriteLine("{0} by {1}", topPlayersScores[i].Item1, topPlayersScores[i].Item2);  
 }  
 }

- В класа „GameEngine“ е въведен конструктор чрез, който ще се извиква метода.

- От класа „GameEngine“ e премахнат „private static int[,] puzzleField“ поради замняната му от обекта „PuzzleField“.

- От класа „GameEngine“ е премахната променливата „rowPositionOfEmptySpace“ поради създаденият обект „PuzzleField“, който поема функциите му.

- От класа „GameEngine“ е премахната променливата „colPositionOfEmptySpace“ поради създаденият обект „PuzzleField“, който поема функциите му.

- Парчето код което отговаря за смяната на стойностите на клетките в полето при придвижване на число е променен от:

int currentlySelectedCell = puzzleField[rowPositionOfTheSelectedNumber, colPositionOfTheSelectedNumber];

puzzleField[rowPositionOfTheSelectedNumber, colPositionOfTheSelectedNumber] =

puzzleField[rowPositionOfEmptySpace, colPositionOfEmptySpace];

puzzleField[rowPositionOfEmptySpace, colPositionOfEmptySpace] = currentlySelectedCell;

rowPositionOfEmptySpace = rowPositionOfTheSelectedNumber;

colPositionOfEmptySpace = colPositionOfTheSelectedNumber;

countOfTotalMoves++;

на  
  
int cellForChange = selectedCell.Context;

selectedCell.Context = this.PuzzleField.EmptyCell.Context;

this.PuzzleField.EmptyCell.Context = cellForChange;

countOfTotalMoves++;

- Кода за проверка дали полето е подредено правилно от метода „IsPuzzleSolved()“ е променено от

private bool IsPuzzleSolved()

{  
 if (puzzleField[3, 3] == 0)

{

int correctNumber = 1;

for (int row = 0; row < 4; row++)

{

for (int col = 0; col < 4; col++)

{

if (correctNumber <= 15)

{

if (puzzleField[row, col] == correctNumber)

{

correctNumber++;

}

else

{

return false;

}

}

else

{

return true;

}

}

}

}

return false;

}

на

private bool IsPuzzleSolved()

{

for (int i = 0; i < this.PuzzleField.Body.Count - 1; i++)

{

Cell currentCell = this.PuzzleField.Body[i];

if (currentCell.Context != i + 1)

{

return false;

}

}

return true;

}

- Премахнат е реда с „i--“ от условните оператори намиращи се в цикъла на метода „Shuffle“ намиращ се в класа „RandomShuffle“ заради случаите, в който никой от условните оператори не успее да направи промяна в пъзел полето се увеличава броя на циклите с 1000+ и така се намалява бързодействието на играта. За това е жертвана част от възможните разбърквания заради оптималност и бързина на играта.

- Кода вътре в метода на RearrangePuzzleField намиращ се в класа „RandomShuffle“ е променен от:  
private void RearrangePuzzleField(PuzzleField puzzleField, Cell selectedCell)

{

Cell emptySpaceCell = puzzleField.EmptyCell;

puzzleField.EmptyCell.Context = selectedCell.Context;

selectedCell.Context = emptySpaceCell.Context;

puzzleField.EmptyCell.Row = selectedCell.Row;

puzzleField.EmptyCell.Col = selectedCell.Col;  
 puzzleField.EmptyCell.Row = selectedCell.Row;

puzzleField.EmptyCell.Col = selectedCell.Col;

}

на

private void RearrangePuzzleField(PuzzleField puzzleField, Cell selectedCell)

{

int index = selectedCell.Col + selectedCell.Row \* puzzleField.MatrixSize;

selectedCell = puzzleField.Body[index];

int emptySpaceCell = puzzleField.EmptyCell.Context;

puzzleField.EmptyCell.Context = selectedCell.Context;

selectedCell.Context = emptySpaceCell;

}

- Частта от кода в условните оператори намиращи се в цикъла на метода „Shuffle“ намиращ се в класа „RandomShuffle“ отговаряща за промяната на позициите на две клетки при разбъркването на полето е отделен в метода „RearrangePuzzleField(puzzleField, selectedCell)“

- Добавени са променливите private int content, private int row, private int col в класа “Cell”.

- Поради добавената имплементация на „Adapter“ патерна кода за принтирането на ранглистата в класа „ConsolePrinter“ и в метода „PrintScoreboard(List<Player> topPlayers)“ се променя от   
  
for (int i = 0; i <= countOfTopPlayers - 1; i++)

{

Console.WriteLine("{0} by {1}", topPlayersScores[i].Name, topPlayersScores[i].TotalMoves);

}

на  
  
for (int i = 0; i <= countOfTopPlayers - 1; i++)

{

Player currentPlayer = new PlayerAdapter();

currentPlayer.Name = topPlayers[i].Name;

currentPlayer.TotalMoves = topPlayers[i].TotalMoves;

currentPlayer.Print();

}

- Полето „int CountTotalMoves“ е премахнато от „GameEngine“ класа поради създаденият обект „Player“, на който се присвоява тази опция.

- Премахнат е параметъра „InitialValue“ от конструктора на “PuzzleField”.

- В класа „PuzzleField“ e добавен метода „PuzzleField GetInstance(int size)“ поради внедряването на Singleton дизайн патерна.

3) Преименувани променливи:

- Променливата “топКандидати” е променена на “topPlayersScores”.

- Променливата „static int broqch“ е променена на „static int countOfTotalMoves“.

- Променени са променливите “int i” на “int row” и “int j” на “int col”, които се намират във вложените цикли в статичният метод Move(int number) и в статичният метод startagain();

- Променена е променливата „string s“ на „string inputCommand“ намираща се в статичният метод startagain() в условната конструкция while (!flagSolved);

- Променена е променливата „int n“ на „int selectedNumber“ намираща се в статичният метод startagain() в условната конструкция while (!flagSolved);

- Променена е променливата „static int [,] a“ на „static int [,] puzzleField“.

- Променена е променливата „bool flag2“ на “bool isTheMoveAreLegal ”, която се намира в метода “MoveTheNumberOfField(int number)”.

- Променена е променливата “bool flagSolved“ на „bool isGameWon“, която се намира в Main().

- Променена е променливата „bool flag“ на „ bool inputIsANumber”, която се намира в Main().

- Променена е променливата „static int x“ на „static int rowPositionOfEmptySpace“, която се намира в началото на кода.

- Променена е променливата „static int y“ на „static int colPositionOfEmptySpace“, която се намира в началото на кода.

- Променена е променливата „static bool flag2“ на „static bool gameContinues“, която се намира в началото на кода.

- Променена е променливата „int k“ на „int rowPositionOfTheSelectedNumber“, която се намира в метода „MoveTheNumberOfField(int number)“.

- Променена е променливата „int l“ на „int colPositionOfTheSelectedNumber“, която се намира в метода „MoveTheNumberOfField(int number)“.

- Променена е променливата „bool flag“ на „bool positionOfNumberIsFound“, която се намира в метода „MoveTheNumberOfField(int number)“.

- Променена е променливата „static int topCount“ на „static int countOfTopPlayers“, която се намира в началото на кода.

- Променена е променливата „int temp“ на „int currentlySelectedCell“, която се намира в метода „MoveTheNumberOfField(int number)“.

- Променена е променливата „int n“ на „int randomNumber“, която се намира в метода „ShuffleThePuzzleField()“.

- Променена е променливата „int nx“ на „int rowOfSelectedCell“, която се намира в метода „ShuffleThePuzzleField()“.

- Променена е променливата „int ny“ на „int colOfSelectedCell“, която се намира в метода „ShuffleThePuzzleField()“.

- Променена е променливата „string s1“ на „string inputOfPlayerName“, която се намира в метода „Main()“.

- Променена е променливата „string res“ на „string resultOfTheGame“, която се намира в метода „Main()“.

4) Въведени константи:

- В класа „GameEngine“ е изведена константата „MatrixSize“ използвана в инстанцията на класа „PuzzleField“.

- В класа „GameEngine“ е изведена константата „InitialValue“ използвана в инстанцията на класа „PuzzleField“.

5) Дизайн патерни:

Prototype design pattern - Creational Pattern

- При запълването на обекта „PuzzleField“ се клонира обекта „Cell“ чрез „Cell currentCell = singleCell.Clone() as Cell“ и след това му се добавят стойности към полетата Context, Row, Col.

Singleton design pattern - Creational Pattern

- Този дизайн патерн е имплементиран в класа „PuzzleField“, за да може да бъде инстанциран само един единствен път. Патерна е направен така, че дори и при многонишково програмиране пак да неможе да се направят повече от една инстанции.

Facade design pattern - Structural Pattern

- „GameEngine“ е ползван като фасада за класа „Program“, за да скрие от него всички останали класове и обекти имплементирани в играта.

Adapter design pattern - Structural Pattern

- Този дизайн патерн е имплементиран да служи като адаптер за класа „Player“, който стандартно няма функционалността да се принтира върху конзолата, а след прилагането на адаптера вече може да принтираме даден играч на конзолата.

Command design pattern - Behavioral Pattern

- В класа „GameEngine“ са добавени своиствата public ICommand TopCommand { get; set; }, public ICommand ExitCommand { get; set; }, public ICommand RestartCommand { get; set; } на който са им зададени стойности чрез метода „DefineCommands“ и който по късно ще бъдат извиквани от класа „CommandManager“.

Strategy design pattern - Behavioral Pattern

- Метода „Shuffle“ е енкапсулиран в класа „RandomShuffle“.