**Структура на проекта: класове, енумерации, конструктори и методи:**

1. **class Board**
   1. public Board(int rows, int columns, int minesCount) – 22 реда (6. София)
   2. public enum Status { SteppedOnAMine, AlreadyOpened, SuccessfullyOpened, AllFieldsAreOpened } – 4 реда (6. София)
   3. private int GenerateRandomNumber(int minValue, int maxValue) – 5 реда (2. Йордан)
   4. private int ScanSurroundingFields(int row, int column) – 57 реда (2. Йордан)
   5. private void SetMines() – 16 реда (2. Йордан)
   6. private bool CheckIfWin() – 21 реда (6. София)
   7. public void PrintGameBoard() – 47 реда (5. Добромир)
   8. public Status OpenField(int row, int column) -29 реда (5. Добромир)
   9. public void PrintAllFields() – 51 реда **(3. Жико)**
   10. public int CountOpenedFields() – 16 реда **(3. Жико)**
2. **class Field**
   1. public Field() – 5 реда (6. София)
   2. public enum FieldStatus { Closed, Opened, IsAMine } – 4 реда (6. София)
3. **class Game**
   1. private static void InitializeGameBoard() – 4 реда (4. Ладислав)
   2. private static void InitializeTopPlayers() – 5 реда (4. Ладислав)
   3. private static bool CheckHighScores(int score) – 17 реда (4. Ладислав)
   4. private static void topadd(ref Player player) – 14 реда (4. Ладислав)
   5. private static void top() – 8 реда (4. Ладислав)
   6. static void Main(string[] args) – 4 реда (4. Ладислав)
   7. private static void Menu() – 112 реда (1. Иржи)
4. **class Player : IComparable**
   1. public Player(string name, int score) – 5 реда (6. София)
   2. public int CompareTo(object obj) – 10 реда (6. София)
   3. public override string ToString() – 6 реда (6. София)

**Общо 462 реда код в методи в методи и конструктори.**

# Documentation for Project “ Minesweeper-5”

Текста в черно е това което е извършено до момента.

Текста в червено е изваден от заданието

1. Renamed the solution to Minesweeper-5.

## Redesigned the project structure:

* Renamed the project to Minesweeper.
* Renamed the main class Program to GameFifteen.
* Extracted each class in a separate file with a good name: GameFifteen.cs, Board.cs, Point.cs.
* …

## Reformatted the source code:

* Removed all unneeded empty lines, e.g. in the method PrintGameBoard()
* Inserted empty lines between the methods.
* Split the lines containing several statements into several simple lines, e.g.:

public enum FieldStatus { Closed, Opened, IsAMine }

to:

public enum FieldStatus

{

Closed, Opened, IsAMine

}

* Formatted the curly braces { and } according to the best practices for the C# language.
* Put { and } after all conditionals and loops (when missing).
* Character casing: variables and fields made camelCase; constants, types and methods made PascalCase.
* Formatted all other elements of the source code according to the best practices introduced in the course “High-Quality Programming Code”.
* Remove unused directives
* Put all using directives inside of the namespace
* Put constructors after fields
* Put private methods after all public methods
* Add 'this.' prefix to indicate that the items are members of the class
* Add access modifiers of all classes and methods
* …

## Renamed variables:

* In class Fifteen: number
* numberOfMoves.
* In Main(string[] args): g
* gameFifteen.

## Introduced constants:

* GAME\_BOARD\_SIZE = 4
* SCORE\_BOARD\_SIZE = 5.

## Extracted the method GenerateRandomGame() from the method Main().

## Introduced class ScoreBoard and moved all related functionality in it.

## Moved method GenerateRandomNumber(int start, int end) to separate class RandomUtils.

## …

## Change if statements in method ScanSurroundingFields () with two nested for cycles

## Extracted the method IsMineInPosition () from the method ScanSurroundingFields ().

## Add Console.Clear to clear the console after every step

1. Extract method PrintMainMenu() from Game.cs and move to Renderer.cs
2. Add method Write in Renderer.cs that print string on console
3. Change methods Console.Writline in Game.cs with Renderer.Write

## If the player has not entered a name, its name “no name”

## .