Refactoring Documentation for Project “Minesweeper - 3”

## TEAM “ASTATINE”

1. Redesigned the project structure:
   * Renamed the project to **Minesweeper**.
   * Renamed the main class **MinesweeperTest** to **Minesweeper**.
   * Extracted each class from **Minichki.cs** in a separate file with a good name: **Field.cs**, **ScoreBoard.cs**, **ConsoleIOManager.cs, Engine.cs**.
   * Added interfaces for the non-static classes for better abstraction – **IField.cs, IScoreBoard.cs.**
   * Class diagram added.
2. Reformatted the source code:
   * Removed all unneeded empty lines in all classes.
   * Inserted empty lines between the methods, after conditional and loop structures and where necessary for better readability and understanding of the code.
   * Put each statement on a several line
   * Formatted the curly braces **{** and **}** according to the best practices for the C# language.
   * Put **{** and **}** after all conditionals and loops (when missing).
   * Indentation improved
3. Renamed identifiers:
   * Casing of names improved – all varaibles renemad using **camelCase;** all methods, classes and constants renamed using **PascalCase.**
   * Identifiers names changed with more descriptive ones, e.g:
     1. In class **ScoreBoard: FirstFive** 🡪 **MaxShowedPlayersCount**
4. Introduced constants:
   * **MaxShowedPlayersCount= 5**
   * **MinesNumber = 15**
   * **MatrixRows = 5**
   * **MatrixCols = 10**
5. **Classes and methods refactored.**
   * **ScoreBoard** class separated and all functionality reletaed to the score board of the game moved in it.
     1. **PrintScoreBoard** method renamed to **Generate** which now returns string to avoid the coupling with the console.
     2. **IScoreBoard** interface added for better abstraction and communication with the other classes and objects.
   * **Field** class separated and all functionality related to the field moved in.
     1. **FillWithRandomMines** method added to the class
     2. The old method **Boom** refactored to have just one goal, renamed to **IsMine** in the **Field** class
     3. Properties added for easier use of the class
     4. **IField** interface added for better abstraction and communication with the other classes and objects.
   * **ConsoleIOManager** class added and all functionality for handling the input and output from the console moved to it.
     1. Each message, printing or getting user data has separate method to call.
     2. Static class for better performance
     3. **IIOManager** interface added for better abstraction and communication with the other classes and objects.
   * **Engine** class separated and all functionality related to executing the main logic of the game moved in it.
     1. **PlayMines** method renamed to **Play.** All ‘goto’ clauses replaced with ‘while’ loops.
     2. **Zapochni** method split to **Initialize** method in the **Field** class and **StartNewGame** in the **Engine**
     3. Validating input row and col moved to separate method **ValidateRowAndCol**
     4. **Proverka** method renamed to **IsCommandEntered**
     5. Added ‘(FormatException fe)’ to the catch clause in the **IsMoveEntered** method
     6. **IEngine** interface added for better abstraction and communication with the other classes and objects.
6. All classes and methods are refactored according to the best practices introduced in the course “[High-Quality Programming Code](http://codecourse.telerik.com/)”.
7. **Unit tests** added for all public members, covering the main and border cases.
8. The **bug** with entering coordinates of already opened cell fixed.
9. XML documentation added for all public members and generated to **Minesweeper.chm** file.