# Sample Refactoring Documentation for Project “Minesweeper-5”

Team “Einsteinium”

1. Redesigned the project structure:
   * Renamed the project to **Minesweeper-5**.
   * …
2. Reformatted the source code:
   * Removed all unneeded empty lines in the project.
   * Added XML documentation.
   * Inserted empty lines between the methods.
   * Cleared most style cop warning.
   * Renamed **Field** class to **FieldCell** for better readability.
   * Removed **GenerateRandomNumber()** method from Game class. Now uses **rand.Next().**
   * Changed logic of **SetMines()** method. Now uses while loop.
   * Catch blocks now handle only expected exceptions.
   * Renamed **CheckHighScore()** method to **IsQualifiedForScoreBoard()** to better reflect what the method does.
   * Changed logic in **CheckIfWin().** Instead of scanning every time the method is called to see which fields are opened, we keep their count using a private variable **openFields** in the class.
   * Removed method **CountOpenedFields()** from **Board** class. Introduced new property to replace it, **OpenedFieldsCount,** which gives read-only access to the variable holding this count.
   * Removed unnecessary cast from **Top()** method.
   * Changed “while” loop from **Menu()**  to “do while” loop. Better suits the purpose.
   * Extracted logic for playing game in **PlayMove()** method**.**
   * Extracted logic for saving high scores **SaveHighScore()** method**.**
   * Extracted logic for processing commands to **ProcessCommands()** method.
   * Merged cases for winning and losing the game, since most of the logic is the same.
   * Removed Field wrapper class and left FieldCell class only.
   * Removed unnecessary **ref** keyword from AddPlayerToScoreBoardMethod()
   * Added a **Console.Clear()** at each loop iteration. Looks better.
   * Extracted logic for **HighScore** to a separate class.
   * Added unit testing project.
   * Split the lines containing several statements into several simple lines, e.g.:

|  |  |  |
| --- | --- | --- |
| **if (input[i] != ' ') break;** | **🡪** | **if (input[i] != ' ')**  **{**  **break;**  **}** |

* + 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**; 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](http://codecourse.telerik.com/)”.

1. Renamed variables:
   * In class **Game**: **choosenRow** 🡪 **chosenRow**.
   * Inclass **Board**: **field** 🡪 **cell**.
   * Inclass **Board**: **fields** 🡪 **field**.
   * In class **Game: str** 🡪 **input**
   * **FieldStatus** 🡪 **CellStatus**
   * **OpenedFields** 🡪 **OpenedCellsCount**
   * **ScanSurroundingFIelds** 🡪**Scan(ScanSurroundingCells())**
   * **mines** 🡪 **minesCount(ScanSurrdoundingCells())**
   * **PrindAllFields()**🡪 **PrintGameboardRevealed()**
2. Introduced constants:
   * **GAME\_BOARD\_SIZE = 4**
   * **SCORE\_BOARD\_SIZE = 5**.
3. Extracted the method **GenerateRandomGame()** from the method **Main()**.
4. Introduced class **ScoreBoard** and moved all related functionality in it.
5. Moved method **GenerateRandomNumber(int start, int end)** to separate class **RandomUtils**.

…