|  |
| --- |
| Refactoring Documentation for Project “Minesweeper – 5”Team "Minesweeper-3"  1. Providing a new Solution folder. 2. Separating the project into three main subprojects.   2.1 Class Library Application  2.2 Console Client Application  2.3 Unit Test Application   1. Dividing the game classes into separate files, stored into the Class Library Application project 2. Separating the static methods into new static class 3. Refactoring MinesInitializer class main methods - > PlayGame method (removing goto command) 4. Providing refactoring and code comments 5. MinesInitializer class refactored according to Singleton design pattern. 6. private static MinesInitializer onlyInstance field contains the only instance of the class 7. private MinesInitializer() disables external instantiation of the class 8. public static MinesInitializer Instance() creates the only instance with lazy loading and returns it to the clients 9. Basic interfaces created    1. IScoreBoard for the score board implementations    2. IMinesGenerator for the random mines generator implementations    3. IMinesGenerator.FillWithRandomMines(…) generates the mines using random deployment algorithm    4. IDrawer for the UI related operations    5. IMinesweeperFactory for the game factory; through this interface implementations of the ones above must be obtained 10. IMinesweeperFactory implements the “Abstract factory” design pattern 11. public class MinesGenerator : IMinesGenerator is the generator of mines used by the game 12. class ScoreBoard : IScoreBoard moved to dedicated folder 13. interface IDrawer defined:     1. void ShowWelcome(string message) displays the game start message and instructions.     2. void ShowGameEnd(string message) displays the game end message.     3. void Draw(string[,] minesField, bool revealMines = false) renders the mines field to the UI     4. void Message(string message) displays message to the user. |
|  |