## Team “Minesweeper-3”

## Participant:

## 1 - *Ivelin Stanchev*

## 2 - *Plamen Stanev*

## 3 - *Stoyan Stoyanov*

## 4 - *Joan Sirakov*

## 5 - *Peter Milchev*

## 6 - *Tancho Mihov*

## 7 - *Iliyan Jordanov*

GitHub repository: <https://github.com/mihov/Minesweeper-3>

Refactoring steps:

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.
14. class ConsoleDrawer : IDrawer introduced. It renders the game output to text console.
15. void PrintInitialMessage() moved to: class InitializerExtensions -> class MinesInitializer
16. public static void InitializerExtensions.Display(string[,] minesMatrix, bool boomed) commented out as unnecessary
17. MinesInitializer.StartPlayCycle() refactored to use IDrawer
18. MinesInitializer.PrintInitialMessage() refactored to use IDrawer
19. Interface IUserInput introduced. It is the generic interface for retrieval of user input.
    1. string GetCommand() returns user input as string.
    2. string GetUserName() returns the user’s name.
20. public class ConsoleInput : IUserInput introduced. It retrieves the user input from text console.
21. MinesInitializer.EnterRowColInput() refactored to use IUserInput
22. InitializerExtensions.StartGame() commented out: method no longer needed. Initialization of class fields moved to MinesInitializer.StartPlayCycle()
23. InitializerExtensions.FillWithRandomMines() commented out: method no longer needed. Its functionality taken over by IMinesGenerator implementations.
24. InitializerExtensions.IsMoveEntered(string line) refactored to InitializerExtensions.IsMoveEntered (string line, ref int row, ref int column)
25. MinesInitializer.EnterRowColInput() updated accordingly.
26. MinesInitializer.EnterRowColInput() refactored to use IDrawer entirely.
27. IUserInput GetCommandProvider() added to IMinesweeperFactory
28. class MinesweeperFactory : IMinesweeperFactory introduced. It is the factory of the game.
29. MinesInitializer.PlayMines() refactored to MinesInitializer.PlayMines(IMinesGenerator minesGenerator, IDrawer drawer, IUserInput userInput, IScoreBoard scoreBoard, Random random)
30. private Random random introduced to MinesInitializer
31. Main() updated to use object factory to create all necessary instances.
32. Random randomMines; deleted from StartPlayCycle()
33. StartPlayCycle() updated to use random field instead
34. MinesGenerator minesGenerator removed from StartPlayCycle()
35. drawer = new ConsoleDrawer() removed
36. this.userInput = new ConsoleInput() removed
37. void PrintScoreBoard(IList<KeyValuePair<int, IList<string>>> highScores) introduced to IDrawer
38. ConsoleDrawer.PrintScoreBoard() implemented
39. ScoreBoard.cs moved to MinesweeperGame.Demo
40. MinesGenerator.cs moved to MinesweeperGame.Demo
41. ScoreBoard.PrintScoreBoard() removed method no longer needed
42. EnterRowColInput(…) updated to use IDrawer.PrintScoreBoard()
43. ScoreBoard.AddPlayer() fix: parameter validation bug
44. EnterRowColInput(…) fix: discard blank user names
45. MediatorExtensions.cs: constants made public
46. MinesGenerator.cs: constant name fix MinesSymbol -> MINES\_SYMBOL
47. Minesinitializer.cs: code clean up
48. EnterRowColInput() renamed to ProcessCommands()
49. ProcessCommands() refactored
50. MoveTo() extracted
51. CheckForGameEnd() renamed to IsValidCommand()
52. enum CommandResult introduced:
    1. ContinueGame – game should continue
    2. RestartGame – start new game
    3. EndApplication – end applicaton
53. StartPlayCycle() refactored to return CommandResult
54. MoveTo() refactored to return CommandResult
55. StartPlayCycle() refactored to return CommandResult
56. MoveTo() documentation enhanced
57. Implementing the **Mediator Pattern**: The MediatorExtensions Class is working together with the MinesInitializer Class the Main game instances during the game play
58. Implementing the **IRepository Pattern**: The Repository Class, implements IRepository interface and provides Get players from the database and Add player methods.
59. The Repository Class is working together with the database file : players.xml