|  |  |  |  |
| --- | --- | --- | --- |
| Sample Refactoring Documentation for Project “Minesweeper” Team “Titanium”   1. Redesigned the project structure:    * Renamed the namespace **Mini** to **Mines**.    * Renamed the main class **Програма** to **MinesweeperGame**.    * Extracted class Boards from MineField    * Renamed other classes (from -> to):      + - **Табло** -> **Board**        - **Команда** -> **Commands**        - **Дъска** -> **MineField**        - **Човек** -> **Player**        - **Command -> CommandParser** 2. Reformatted the source code:  * Removed all unneeded empty lines, in the methods * Removed unnecessary comments in Latin or Cyrillic. * Removed all magic numbers and replaced them either with constants or default values; * Removed all unused usings. * Added this. , where it’s necessary. * Updates in class CommandParser:   1. Changed the class to non-static   2. Renamed the class Command to CommandParser   3. Renamed Method Прочети() to ReadCommand();   4. Renamed Method Clear() to CommandsInitialization();   5. Put a new lines to separate the program logic blocks;   6. Renamed InvalidCommand to ValidCommand   7. Thus update !InvalidCommand -> ValidCommand when needed   8. Renamed ReadCommand() to TryParseCommand()   9. Added IsSpecialCommand()   10. Extracted ParseCoordinates() from NexMove()   11. Removed Boolean fields (valid command, restart, exit, top)   12. Added private field command; * Updates in class MineField:  1. Renamed Display() to ToString() – replaced the Console.WriteLine with a StringBuilder that gathers the information about the current state of the board and returns it as string. 2. Renamed Proverka1() to IsInsideTheField(); - Split the complicated Boolean expressions to easy-to-understand expressions; 3. Renamed Proverka2() to IsAlreadyShown(); 4. Renamed Proverka3() to IsMine(); 5. Renamed Край() to RevealAllMines() – changed the iteration logic to for() cycle that works for field of any size. 6. Extracted method ActualizeNeighbours() from PlaceMine() 7. Extracted boards class that holds the information about the boards used in MineField class. The class is internal for the game project. 8. Renamed the arrays that hold the information of the board (shown -> IsShown, number -> NumberOfNeighbourMines, etc.) 9. Extracted all game messages to static class GameMessages. 10. Added Exceptions to the class constructor. 11. Added AddRemoveFlag() with new game functionality for adding and removing flags  * Updates to class MineSweeperGame  1. Removed all goto statements and replaced them with proper action that handles the game restart command (e.g. calling the Main()) 2. Extracted ExecuteCommand() from Main() 3. Extracted ReadCommand() from Main() 4. Extracted ExecuteSpecialCommand() from ExecuteCommand() – that handles all special commands in the game e.g. (restart, exit, flag etc.) 5. Extracted ExecuteRevealBlockCommand() from ExecuteCommand(); 6. Renamed board to minefield 7. Extracted scoreboard field out of Main() as a static field to prevent the data losses when game is restarted. 8. Replaced the switch statement in ExecuteRevealBlockCommand() with if statements. 9. Added ExecuteFlagCommand() that execute flag / unflag commands. 10. Added shouldDisplayBoard Boolean field;  * Inserted empty lines between the methods. * 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). * Renamed variables from Latin-Bulgarian to English. * 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/)”. |