## “Iridium” Team

# “King Survival” Refactoring Log

### Data Level Refactoring

1. The organization [**IridiumTeam**](https://github.com/IridiumTeam) is created at [GitHub](https://github.com/) and a [**KingSurvival**](https://github.com/IridiumTeam/KingSurvival)repository is added.
2. The solution is renamed to *KingSurvival.sln.*
3. The project file is renamed to *KingSurvival.csproj.*
4. The source code in the *KingSurvival.cs* file is reformatted and the empty lines – removed.
5. All comments are removed.
6. Proper naming of all members is done.

|  |  |
| --- | --- |
| Old member | New member |
| private bool proverka(int row, int colum) | private bool IsPositionOnTheChessboard(int row, int col) |
| private bool proverka2(int row, int colum) | private bool IsPositionValid(int row, int col) |
| public bool KingLost() | public bool KingLoses() |
| public bool KingWon() | public bool KingWins() |
| private int[,] dyska; | private bool[,] occupied; |
| public void PrintBoard() | public override string ToString() |
| private int whiteCell = '+'; | private const char WhiteSquareCharacter = '+'; |
| private int blackCell = '-'; | private const char BlackSquareCharacter = '-'; |
| public bool MoveKingIfPossible(string command) | public bool TryMoveKing(string command) |
| public bool MovePawnIfPossible(string command) | public bool TryMovePawn(string command) |

1. Conversion of some primitive data members to a class.

|  |  |
| --- | --- |
| Old member | New member |
| private int[] deltaRed = { -1, +1, +1, -1 }; | Class Move |
| private int[] deltaColona = { +1, +1, -1, -1 }; |  |

1. Unused members are removed.

|  |  |
| --- | --- |
| Old member | New member |
| private int[] pawnRows = { 0, 0, 0, 0 }; |  |
| private int[] pawnColumns = { 0, 2, 4, 6 }; |  |
| private int kingRow = 7; |  |
| private int kingColumn = 3; |  |
| public void DaiMiDyskata() |  |

### Method Level and Class Level Refactoring

1. Methods of the **KingSurvival.cs** file are transferred to the **ChessboardManager.cs** class.
2. Introduced new object-oriented design with several classes:

* [**ChessPiece.cs**](https://github.com/IridiumTeam/KingSurvival/blob/master/Source/KingSurvival/ChessPiece.cs)
* [**ChessPieceType.cs**](https://github.com/IridiumTeam/KingSurvival/blob/master/Source/KingSurvival/ChessPieceType.cs)
* [**ChessboardManager.cs**](https://github.com/IridiumTeam/KingSurvival/blob/master/Source/KingSurvival/ChessboardManager.cs)
* [**Game.cs**](https://github.com/IridiumTeam/KingSurvival/blob/master/Source/KingSurvival/Game.cs)
* [**Move.cs**](https://github.com/IridiumTeam/KingSurvival/blob/master/Source/KingSurvival/Move.cs)

1. Classes are implemented and unit tests are run. [**JetBrains dotCover**](http://www.jetbrains.com/dotcover/) is used to measure the code coverage.
2. A documentation file is generated using Sandcastle.