

"King Survival" Refactoring Log

Data Level Refactoring

1. The organization **IridiumTeam** is created at GitHub and a **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
<code>private bool proverka(int row, int colum)</code>	<code>private bool IsPositionOnTheChessboard(int row, int col)</code>
<code>private bool proverka2(int row, int colum)</code>	<code>private bool IsPositionValid(int row, int col)</code>
<code>public bool KingLost()</code>	<code>public bool KingLoses()</code>
<code>public bool KingWon()</code>	<code>public bool KingWins()</code>
<code>private int[,] dyska;</code>	<code>private bool[,] occupied;</code>
<code>public void PrintBoard()</code>	<code>public override string ToString()</code>
<code>private int whiteCell = '+';</code>	<code>private const char WhiteSquareCharacter = '+';</code>
<code>private int blackCell = '-';</code>	<code>private const char BlackSquareCharacter = '-';</code>
<code>public bool MoveKingIfPossible(string command)</code>	<code>public bool TryMoveKing(string command)</code>
<code>public bool MovePawnIfPossible(string command)</code>	<code>public bool TryMovePawn(string command)</code>

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

Old member	New member
<pre>private int[] deltaRed = { -1, +1, +1, -1 };</pre>	Class Move
<pre>private int[] deltaColona = { +1, +1, -1, -1 };</pre>	

8. Unused members are removed.

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

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**
 - **ChessPieceType.cs**
 - **ChessboardManager.cs**
 - **Game.cs**
 - **Move.cs**
3. Classes are implemented and unit tests are run. **JetBrains dotCover** is used to measure the code coverage.
4. A documentation file is generated using Sandcastle.