# Refactoring Documentation for Project “Hangman-2”

***Team “Technetium”***

1. Redesigned the project structure:

* Renamed the project to form KPK.Proekt1 to Hangman
  + Renamed the main class Program to GameFifteen.
  + Extracted class PlayerMistakes from CommandExecutor class in a separate class: PlayerMistakes.

1. Reformatted the source code:
   * Removed all unneeded empty lines, e.g. in the class CommandExecutor
   * Inserted empty lines between the methods.
   * Split the lines containing several statements into several simple lines, e.g.:

|  |  |  |
| --- | --- | --- |
| for (int i = 0; i < 5; i++)  if(scoreboard[i] != null)  Console.WriteLine("{0}. {1} ---> {2}", i+1, scoreboard[i].PlayerName, scoreboard[i].NumberOfMistakes); | 🡪 | for (int i = 0; i < 5; i++)  {  if (Scoreboard[i] != null)  {  Console.WriteLine("{0}.{1} ---> {2}",  i + 1, Scoreboard[i].PlayerName, Scoreboard[i].NumberOfMistakes);  }  }  Console.WriteLine(); |

* + Formatted the curly braces { and } according to the best practices for the C# language.
  + Put { and } after all conditionals and loops (when missing).
  + 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/)”.
  + …

1. Renamed variables:
   * In class CommandExecutor: scoreboard 🡪 Scoreboard due to it is public access.
   * In Main(string[] args): g 🡪 gameFifteen.
   * …
2. Introduced constants:
   * HIGH-SCORE-NUMBER=5;
   * …
3. Extracted the methods: (example GenerateRandomGame() from the method Main().)
4. Introduced classes: (example ScoreBoard and moved all related functionality in it).

* Create new class PlayersScore;
* Move there method ScoreSort from WordInitializator class;
* Move Scoreboard field from Command Executor class;
* Move public static void TopResults() from CommandExecutor;

1. Moved method to separate class (method GenerateRandomNumber(int start, int end) to RandomUtils.)
2. Encapsulate fields in all classes;