Sample refactoring documentation for the Hangman-1 project

1. Redesigned the project structure

* In class ScoreBoard: NUMBER\_OF\_SCORES constant to private
* Added new class library to implement the main logic
* Added interfaces IWordsContainer, IGameEngine, IMessageProvider, IParser, IReader, IScoreBoard
* Added abstract classes Renderer, GameEngineFactory
* Added classes to implement the functionality ConsoleRenderer, DefautGameEngineFactory, ConsoleReader, DefaultParser, DefaultMessageProvider, DefaultScoreboard, GameEngine, WordsContainer
* Added RandomGenerator class that is singleton design pattern

1. Refactored the source code

* Added new line everywhere after an if-else and loop blocks
* Removed unnecessary comments in the Hangman class
* Removed unnecessary empty lines in the Hangman class
* Set spacing between methods to be single empty line
* Moved the words string array and random generator fields to the top
* Arranged every if-else block body to be under the if-else statements

1. Renamed variables

* In class ScoreBoard: mistackes array -> mistakesCollection
* In class ScoreBoard: mistacke -> mistake
* In class ScoreBoard: Hangman class -> HangmanMain
* In class ScoreBoard: besenica class -> Hangman
* In class Hangman: Mistacke property -> Mistake