### Refactoring Documentation for Project "Poker"

1. Redesigned the project structure:
   * Renamed the main class **Program** to **PokerMain**.
   * Renamed the **Form1** class to **PokerTable**.
   * Extracted the **player** and **bots logic** into separate classes **BotPlayer.cs, HumanPlayer.cs**, **Player.cs** in order to achieve **strong cohesion** and **single responsinility**.
2. Reformatted the source code:
   * Added a lot of empty lines in the class PokerTable.cs to improve code readability.
   * Inserted empty lines between the methods.
   * Split the lines containing several statements into several simple lines.

* 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 most of the elements of the source code according to the best practices introduced in the course "High-Quality Programming Code". Were unable to make all elements quality due to the extremely low quality and high level of unreadability of the initially provided source code.
* Managed to rename with meaningfull names most of the variables and methods in the code, which was not an easy task, given the senseless naming of the initially provided code.
* Changed and applied appropriate access modifiers to variables and methods.

1. Renamed variables:
   * In class **PokerTable:** 
     1. **pPanel** ➔ **playerPanel**.
     2. **b1Panel** ➔ **bot1Panel**.
     3. **b2Panel** ➔ **bot2Panel**.
     4. **b3Panel** ➔ **bot3Panel**.
     5. **b4Panel** ➔ **bot4Panel**.
     6. **b5Panel** ➔ **bot5Panel**.
     7. **bools** ➔ **bankruptPlayers**.
     8. **Win** ➔ **winningHands**.
     9. **Deck** ➔ **deckCardsImages**.
     10. **Holder** ➔ **pictureBoxDeckCards**.
     11. **call** ➔ **bigBlindValue**.
     12. **width** ➔ **windowWidth**.
     13. **height** ➔ **windowHeight**.
     14. **and many others**.
   * A lot of the variables with meaning less names that concerned the bots and player logic were transformed to respective properties in the bots and player classes.
2. Introduced constants:
   * In class **PokerTable:** 
     1. InitialNumberOfBots = 5;
     2. DefaultBigBlindValue = 500;
     3. DefaultSmallBlindValue = 250;
     4. DefaultChipsCount = 10000;
     5. NumberOfCardsInDeck = 52;
     6. TableCardsCount = 17
3. Extracted the following methods to make the logic in the PokerTable.cs class more readable:
   * InitializePlayer().
   * InitializeBots().
   * DisableTextBoxesUserInteraction().
   * InitializeTextBoxes().
   * SetBlindButtonsVisibilityToFalse().
4. Extracted the following methods to a new class **WinnigHands.cs** to make the logic in the PokerTable.cs class more readable and in an effort to achieve **strong cohesion** and **single responsibility**:
   * RulesStraightFlush().
   * RulesFourOfAKind().
   * RulesFullHouse().
   * RulesFlush().
   * RulseStraight().
   * RulesThreeOfAKind().
   * RulesTwoPair().
   * RukesPairTwoPair().
   * RulesPairFromHand().
   * RukesHighCard().