|  |
| --- |
| Sample Refactoring Documentation for Project "Poker"  1. Redesigned the project structure:    * Renamed the main class **Program.cs** to **PokerGame.cs**.    * Renamed the class **Form1.cs** to **PokerGameEngine**.cs    * Extracted the following from **Form1.cs** to separate namespaces and files with a good names:      + Namespace: **Core**   Class: **PokerGameEngine.cs, AddChipsForm.cs, Dealer.cs**, **HandEvaluator.cs**,  **ArtificialInteligence.cs,**  Subnamespace: **Database**  Class: **PokerDatabase**.cs  Subnamespace: **Factories**  Class: **PlayerFactory.cs**   * + - Namespace: **Models**   Classes: **Pot.cs, Card.cs, ChipsSet.cs, Deck.cs, Hand.cs,**  **Result.cs , Blind.cs**  Subnamespace: **Players**  Classes**:** abstract **Player.cs, Bot.cs, Human.cs**  Subnamespace: **Enums**  Classes: **CardsRank.cs, Suits.cs, Actions.cs, WinningHandsTypes.cs  Subnamespace: PokerActions** Classes:abstract **Action, AllInAction, CallAction, CheckAction, FoldAction, RaiseAction.cs**   * + - Namespace: **Interfaces**   Classes: **ICard, IChipsSet, IDeck, IHand, IMessagePrintable, IPlayer, IPlayerFactory, IPokerAction, IPokerDatabase, IPot, IResult, IUserInterface, IBlind.**   * + - Namespace: **Constants**  Classes: **Common**.**cs**, **Locations**.**cs** , **Messages**.cs     - Namespace: **IU**   Classes: **WindowsFormUserInterface.cs**   1. Reformatted the source code:    * Remove unused global variables and made the rest private.    * Removed all static fields in Form1 used as ref types and replaced them with objects’ properties.    * Removed all unneeded empty lines in all classes.    * 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 all other elements of the source code according to the best practices introduced in the course "High-Quality Programming Code".  1. Introduced Namespace **Constants** with **Common**.cs, **Locations** and **Messages** classes which hold all constants values. 2. Introduced interface **IDeck** and class **Deck.cs** …… 3. Introduced interface **IHand** and class **Hand.cs** ….. 4. Introduced interface **ICard** and class **Card.cs….** 5. Introduced interface **IPlayer** and abstract class **Player.cs** (with inheritors **Bot.cs** and **Human.cs)** and moved all related functionality in it and remove all global variables concerning players. Used Players properties instead. 6. Introduced interface **IPlayerFactory** and class **PlayerFactory.cs** for creating instances of Player implemented - *Factory* design pattern implemented. 7. Introduced interface **IPokerDatabase** and **PokerDatabase**.cs for adding new Players in the database and gets the array of Players. 8. Introduced interface IPot and class **Pot.cs** , *Singleton* Design pattern implemented. Deleted Pot’s global variables and put them as properties in it. 9. Introduced class Interface **IChipsSet** andclass **ChipsSet.cs** and used as property in Pot and Player. 10. Moved method **Rules(….)** to separate class called **HandEvaluator.cs** with public method Apply and implemented new logic for evalutaion of hands. 11. Moved method **Winner**(..) to separate class **Dealer.cs** with public methods **CheckWinners**() and **DistributePot**(). 12. Renamed class **Type.cs** to **Result.cs**. and added it as property in Player. 13. Moved method **AI(..)** to separate class called **ArtificialIntelligence**.cs with public method **Apply(..)** and private methods changed from e.g. **Straight**(..) to **ActStraight**(..) 14. Created abstract class **Action** with inheritors **AllInAction, CallAction, FoldAction, CheckAction, RaiseAction** 15. Introduced **WindowsFormUserInterface** which implements **IUserInterface** which extends **IPrintMessagable** 16. Create class and interface for **Blinds**. Implemented blinds logic. 17. Chenge **AddChips**, **Small Blind, BigBlind, Raise** textboxes to NumericUpDown. |
|  |
|  |