Team Bulls and Cows – 4

*project documentation*

1. Redesigned the project structure:
   1. Split the main class **bikove\_i\_kravi.cs** into:
      1. **BullsAndCows.cs** – The entry point of the program
      2. **GameEngine.cs** – Here is the main logic of the program
      3. **GameNumberProvider.cs** – A class to provide functionality concerning the
      4. **Helper.cs** -
      5. **InputOutput.cs** -
      6. **MessageDispatcher.cs** -
   2. Introduced interfaces:
      1. **ICalculateBullsAndCowsStrategy** -
      2. **IGameEngine** -
      3. **IMessageDispatcher** -
      4. **IPrinter** -
      5. **IReader** -
   3. Introduced classes:
      1. **Command.cs** – A base class for all the commands.
      2. **CommandCreator.cs** – A factory class to provide new instances of all the command classes.
      3. **CommandExit.cs** – The command to be executed when the player enters ‘exit’, it stops the game and displays the ‘goodbye’ message.
      4. **CommandHelp.cs** – The command to be executed when the player enters ‘help’, it will display a hint of how the actual number looks like.
      5. **CommandOther.cs** - The command to be executed when the player enters an invalid command string, it will display a warning message to the user.
      6. **CommandRestart.cs** - The command to be executed when the player enters ‘restart’, it will restart the game.
      7. **CommandTop.cs** - The command to be executed when the player enters ‘top’, it will display the leader board.
      8. **ConsolePrinter.cs** – A class used to print messages on the console.
      9. **ConsoleReader.cs** – A class used to retrieve user input from the console.
      10. **FactoryMethod.cs** – A base class for the CommandCreator class. Provides an abstract definition of the Create method.
      11. **GameEngine.cs** – Contains all of the main logic of the game.
      12. **GameNumberProvider.cs** – Provides random four digit numbers for the player to guess. Also provides functionality to check if the player entered a valid guess number.
      13. **Helper.cs** -
      14. **InputOutput.cs** -
      15. **LeaderBoard.cs** -
      16. **MessageDispatcher.**cs -
      17. **NormalCalculateBullsAndCowsStrategy.cs** -
2. Reformatted the source code:
   1. Removed all unneeded empty lines
   2. Inserted empty lines between the methods
3. Renamed variables:
   1. **WELCOME\_MESSAGE** to **WelcomeMessage**
   2. **WRONG\_COMMAND\_MESSAGE** to **WrongCommandMessage**
   3. **Klasirane** to **leaderboard**
   4. **maxCountOfStoredDate** to **DefaultNumberOfItemsInLeaderBoard**
4. Renamed methods:
   1. **generateNumber()** to **Generate()**
   2. **PokajiHelp()** to **GetHelp()**
   3. **generateHelpPattern()** to **GenerateHelpPattern()**
   4. **FinishGame()** to **Exit()**
5. Renamed classes:
   1. **klasirane.cs** to **LeaderBoard.cs**
   2. **bikove\_i\_kravi.cs** to **BullsAndCows.cs**
6. Introduced constants:
   1. **DefaultNumberLength** in **GameEngine.cs**
   2. **DefaultNumberOfItems** in **LeaderBoard.cs**
   3. **GoodByeMessage** in **MessageDispatcher.cs**
   4. **InvalidCommandMessage** in **MessageDispatcher.cs**
   5. **NoCheatersMessage** in **MessageDispatcher.cs**
   6. **EnterNameMessage** in **MessageDispatcher.cs**
   7. **EnterCommandMessage** in **MessageDispatcher.cs**
   8. **WelcomeMessage** in **MessageDispatcher.cs**
7. Implemented design patterns:
   1. Factory Method
   2. Command pattern
   3. Strategy pattern
   4. Bridge pattern
   5. Singleton pattern
   6. Facade pattern