Refactoring Documentation

Project “Bulls and Cows”

7/1/2011

**Author:** Anton Angelov

**University:** TU-Sofia

**f.n.** 121208260

* namespace bikove – removed
* unused using removed
* string[] args removed from Main()
* ScoreBoard class file renamed to Scoreboard
* rrr in BullsAndCowsNumber renamed to randomGenerator
* method GenerateRandomNumbers renamed to GenerateRandomDigits
* numberStringBuilder renamed to guessNumber
* cheats property renamed to Cheats
* rezultat renamed to Result
* created constructor for Result struct
* ime renamed to name
* gameScore class renamed to GameScore
* ScoresFile renamed to SCORESBOARD\_FILENAME
* WelcomeMessage renamed to WELCOME\_MSG
* WrongNumberMessage renamed to WRONG\_NUMBER\_MSG
* InvalidCommandMessage renamed to INVALID\_COMMAND\_MSG
* NumberGuessedWithoutCheats renamed to NUMBER\_GUESSED\_WITHOUT\_CHEATS\_MSG
* NumberGuessedWithCheats renamed to NUMBER\_GUESSED\_WITH\_CHEATS\_MSG
* GoodBuyMessage renamed to GOOD\_BYE\_MSG
* Unnecessary comments removed
* scores.txt renamed to Scoreboard.txt
* formatted string constants to fit in one screen

public const string ScoresFile = "scores.txt";

public const string WelcomeMessage = "Welcome to “Bulls and Cows” game. Please try to guess my secret 4-digit number.\nUse 'top' to view the top scoreboard, 'restart' to start a new game and 'help' to cheat and 'exit' to quit the game.";

public const string WrongNumberMessage = "Wrong number!";

public const string InvalidCommandMessage = "Incorrect guess or command!";

public const string NumberGuessedWithoutCheats = "Congratulations! You guessed the secret number in {0} {1}.\nPlease enter your name for the top scoreboard: ";

public const string NumberGuessedWithCheats = "Congratulations! You guessed the secret number in {0} {1} and {2} {3}.\nYou are not allowed to enter the top scoreboard.";

public const string GoodBuyMessage = "Good bye!";

public const string SCOREBOARD\_FILENAME = "Scoreboard.txt";

public const string WELCOME\_MSG =

"Welcome to “Bulls and Cows” game."+

"Please try to guess my secret 4-digit number.\n"+

"Use 'top' to view the top scoreboard, 'restart' to start a new game and 'help'"+

" to cheat and 'exit' to quit the game.";

public const string WRONG\_NUMBER\_MSG = "Wrong number!";

public const string GOOD\_BYE\_MSG = "Good bye!";

public const string INVALID\_COMMAND\_MSG = "Incorrect guess or command!";

public const string NUMBER\_GUESSED\_WITHOUT\_CHEATS\_MSG =

"Congratulations! You guessed the secret number in {0} {1}.\n"+

"Please enter your name for the top scoreboard: ";

public const string NUMBER\_GUESSED\_WITH\_CHEATS\_MSG =

"Congratulations! You guessed the secret number in {0} {1}"+

" and {2} {3}.\nYou are not allowed to enter the top scoreboard.";

* Extracted new method PlayGame() from Main()
* Program file and class renamed to GameBullAndCows
* Create properties for fields in Result structure and delete fields
* MaxPlayersToShowInScoreboard renamed to MAX\_PLAYERS\_COUNT\_IN\_SCOREBOARD
* scores renamed to scoreboard
* switch places of fields and constants
* GameScore class and file renamed to PlayerInfo
* Scoreboard type changed to List
* Format add to scoreboard:

string scoreString = inputStream.ReadLine();

this.scores.Add(gameScore.Deserialize(scoreString));

string scoreString = inputStream.ReadLine();

PlayerInfo currentGameScore = PlayerInfo.Deserialize(scoreString);

string currentName = currentGameScore.Name;

int currentNumberOfGuesses = currentGameScore.Guesses;

this.scoreboard.Add(currentNumberOfGuesses,currentName);

* Parameters of AddScore method changed to PlayerInfo object
* Result changed from struct to class
* Extracted new method InitializeScoreboard(filename);
* SaveToFile method removed
* InitializeScoreboard removed
* Serialize method removed
* GetHashCode of BullsAndCowsNumber removed
* GetHashCode of PlayerInfo removed
* MAX\_PLAYERS\_COUNT\_IN\_SCOREBOARD change to 5
* Fix bugs and format method ToString() of Scoreboard class. Before it could not save players with same guesses count. Showed 11 players instead of 5.

public override string ToString()

{

if (scores.Count == 0)

{

return "Top scoreboard is empty." + Environment.NewLine;

}

StringBuilder scoreBoard = new StringBuilder();

scoreBoard.AppendLine("Scoreboard:");

}

int count = 0;

foreach (gameScore gameScore in scores)

{

count++;

scoreBoard.AppendLine(string.Format("{0}. {1}", count, gameScore));

if (count > MaxPlayersToShowInScoreboard) break;

}

return scoreBoard.ToString();

* Extracted new method CreateScoreboardText()

private string CreateScoreboardText()

{

StringBuilder scoreBoard = new StringBuilder();

scoreBoard.AppendLine("Scoreboard:");

int count = 0;

scoreboard.Sort();

foreach (PlayerInfo currentPlayer in scoreboard)

{

count++;

scoreBoard.AppendLine(string.Format("{0}. {1}", count, currentPlayer));

if (count == MAX\_PLAYERS\_COUNT\_IN\_SCOREBOARD)

{

break;

}

}

return scoreBoard.ToString();

}

int currentNumberOfGuesses = currentGameScore.Guesses;

this.scoreboard.Add(currentNumberOfGuesses,currentName);

* Equals methods removed
* PlayerInfo : Icomparable changed to PlayerInfo : IComparable<PlayerInfo>
* CompareTo() method changed:

public int CompareTo(object obj)

{

gameScore objectToCompare = obj as gameScore;

if (objectToCompare == null)

{

return -1;

}

if (this.Guesses.CompareTo(objectToCompare.Guesses) == 0)

{

return this.Name.CompareTo(objectToCompare.Name);

}

else

{

return this.Guesses.CompareTo(objectToCompare.Guesses);

}

}

public int CompareTo(PlayerInfo otherPlayer)

{

if (this.Guesses.CompareTo(otherPlayer.Guesses) == 0)

{

return this.Name.CompareTo(otherPlayer.Name);

}

else

{

return this.Guesses.CompareTo(otherPlayer.Guesses);

}

}

* string filename parameter of Scoreboard’s constructor removed
* SCOREBOARD\_FILENAME constant removed
* Moved properties before constructor, change order of fields and properties in BullsAndCowsNumber
* Merge the both methods TryToGuess in one single method
* TryToGuess method renamed to FindBullsAndCowsCount
* Moved PlayGame method to BullsAndCowsNumber class
* Moved all constants from BullsAndCowsGame to BullsAndCowsNumber class
* Playgame renamed to Play
* WRONG\_NUMBER\_MSG renamed to INVALID\_NUMBER\_MSG and message changed
* Make unit test for FindCountOfBullsAndCows
* Fix bug when enter spaces before or after guess number.

string numberText = userGuessNumberText.Trim();

* Added unit test for GenerateRandomDigits
* Added property for cheatNumber array
* GameBullAndCows renamed to BullsAndCowsGameStarter
* BullsAndCowsNumber renamed to BullsAndCowsGame
* Change “\n” in string constants with Environment.NewLine
* Added messages if test fails
* Extract method for easily set of the digits of secret number
* Introduce a new static filed NEW\_LINE
* Made const messages readonly
* All fields set to private
* Fix bug in counting bulls and cows
* Introduce variable before return

string cheatNumber = new String(CheatNumber);

* Extract method ExtractGuessNumberDigits
* One field array from digit of the generated number than four different fields

public int FirstDigit { get; private set; }

public int SecondDigit { get; private set; }

public int ThirdDigit { get; private set; }

public int FourthDigit { get; private set; }

public int[] NumberForGuessDigits { get; private set; }

* The code from FindBullsAndCowsCount divided in 3 different methods ExtractGuessNumberDigits, CountBulls, CountCows.
* Multiple if conditions replaced with for cycles
* Introduce const GUESS\_NUMBER\_LENGHT and replace all fours in the code
* Added unit tests for Result
* Added unit tests for PlayerInfo
* Refactored method Play() divided in methods – ReadUserCommand, ExecuteCommand, RestartGame, DisplayCommandResult, AddPlayerToScoreboard, GetPluralOrSingularOfWord
* Added some logic to AddPlayerToScoreboard in BullsAndCowsGame ->

string name = String.Empty;

while(name != String.Empty)

{

Console.Write("Please enter your name for the top scoreboard: ");

name = Console.ReadLine();

if(name == String.Empty)

{

Console.WriteLine("You name for the scoreboard should"+

" consists from at least 1 symblol! Try again!");

}

}

* Extracted result string of ToString method of Scoreboard class

if (scoreboard.Count == 0)

{

string emptyResult = "Top scoreboard is empty." +

Environment.NewLine;

return emptyResult;

}

* Added unit tests for Scoreboard class
* AddScore renamed to AddNewResult in class Scoreboard
* Add unit tests for RestartGame method in BullsAndCowsGame class
* Remove parameter from Restart method in BullsAndCowsGame class
* Add unit tests for ReadUserInput method in BullsAndCowsGame class
* Move AddPlayerToScoreboard to Scoreboard class and change parameters from

Scoreboard and BullsAndCowsGame to int guessesCount. Remove static.

* Added unit test for GetPluralOrSingularOfWord in BullsAndCowsGame class
* Added unit tests for ExecuteCommand method
* Remove unused messages in exception in FindBullsAndCowsMethod
* Remove catch Format exception in ExecuteCommand -> unused
* Remove INVALID\_NUMBER\_MSG
* Make methods Play and ExecuteCommand testable, remove parameter BullsAndCowsGame and change it inside with this
* Change ToString method of Result ->

Wrong number! Bulls: {0}, Cows: {1}

* Make Play() method testable added. Remove case “exit” in ExecuteCommand

if(command == "exit")

{

Console.WriteLine(GOOD\_BYE\_MSG);

return;

}

* In the ExecuteCommand method change catch Exception and print its message
* Remove BullsAndCowsGame parameter in DisplayCommandResult method
* DisplayCommandResult renamed to DisplayGuessResult
* Added unit tests for DisplayGuessResult and fix little bugs in formatting
* Put all class in one global namespace for the project
* Extracted new method ExecuteDefaultCommand
* Removed Restart method
* Refacture DisplayWelcomeMessage. Moved in one WriteLine
* Extracted new method RevealChosenDigit
* AddPlayerToScoreboard a WriteLine replaced with Environment.NewLine
* Extracted new method DisplayResultWithCheats

private void DisplayResultWithCheats()

{

string attemptText = GetPluralOrSingularOfWord(this.GuessesCount, "attempt");

string cheatText = GetPluralOrSingularOfWord(this.Cheats, "cheat");

Console.WriteLine(NUMBER\_GUESSED\_WITH\_CHEATS\_MSG, this.GuessesCount,

attemptText, this.Cheats, cheatText);

}

* Extracted new method DisplayResultWithoutCheats

private void DisplayResultWithoutCheats(Scoreboard scoreBoard)

{

string attemptStr = GetPluralOrSingularOfWord(this.GuessesCount, "attempt");

Console.WriteLine(NUMBER\_GUESSED\_WITHOUT\_CHEATS\_MSG, this.GuessesCount, attemptStr);

scoreBoard.AddPlayerToScoreboard(this.GuessesCount);

}

* Method DisplayGuessResult renamed to DisplayCommandResult

|  |  |
| --- | --- |
| **Legend** | |
|  | Removed |
|  | Extracted or moved |
|  | Renamed |
|  | Changed logic and testing |