| scoreboard |  |
| --- | --- |
| - totalGames: integer - totalWins: integer - totalLoss: integer - gamesEO: integer - winsEO: integer - lossEO: integer  - gamesThimble: integer - winsThimble: integer - lossThimble: integer | - gamesGuess: integer - winsGuess: integer - lossGuess: integer - gamesThread: integer - winsThread: integer - lossThread: integer - gamesToss: integer - winsToss: integer - lossToss: integer |
| + scoreboard( ): Instructor method. Creates a scoreboard object. + printScore(scoreboard): Prints out a detailed breakdown of the scoreboard object passed to the console. + getTotal( ): Returns an integer value stored in totalGames. + getTotalWon( ): Returns an integer value stored in totalWins. + getTotalLoss( ): Returns an integer value stored in totalLoss. + setThread(int): Increments the totalGames and gamesThread by the integer value passed to the method. + setThreadWins(int): Increments the totalWins and winsThread by the integer value passed to the method. + setThreadLoss(int): Increments the totalLoss and lossThread by the integer value passed to the method. + getThread( ): Returns the total games played count for Find the Red Thread. + getThreadWins( ): Returns the total games won count for Find the Red Thread.  + getThreadLoss( ): Returns the total games lost count for Find the Red Thread. + setToss(int): Increments the total games played, and the games played count for Coin Toss by the valued passed. + setTossWins(int): Increments the total games won, and the games won count for Coin Toss by the value passed. + setTossLoss(int)L Increments the total games lost and the games lost count for Coin Toss by the value passed. + getToss( ): Returns the total games played count of Coin Toss. + getTossWins( ): Returns the total games won count of Coin Toss. + getTossLoss( ): Returns the total games lost count of Coin Toss. + getEOLoss( ): Returns the total games lost for Even & Odd. | + setThimble(int): Increments the totalGames and gamesThimble by the integer value passed to the method. + setThimbleWins(int): Increments the totalWins and winsThimble by the integer value passed to the method. + setThimbleLoss(int): Increments the totalLoss and lossThimble by the integer value passed to the method. + getThimble( ): Returns the total games played count for Find The Thimble. + getThimbleWins( ): Returns the total games won count for Find the Thimble. + getThimbleLoss( ): Returns the total games lost count for Find the Thimble. + setGuess(int): Increments the total games played count as well as the total games played for Guess The Number. + setGuessWins(int): Increments the total games won count and the games won count for Guess the Number. + setGuessLoss(int): Increments the total games lost count, and the games lost count for Guess the Number. + getGuess( ): Returns the total games played count for Guess the Number. + getGuessWins( ): Returns the total games won count for Guess the Number. + getGuessLoss( ): Returns the total games lost count for Guess the Number. + setEO(int): Increments the total games played count and the games played count for Even & Odd by the value passed. + setEOWins(int) Increments the total games won and the games won counts for Even & Odd by the value passed. + setEOLoss(int): Increments the total games lost and the games lost counts for Even & Odd by the value passed. + getEO( ): Returns the total games played count for Even & Odd. + getEOWins( ): Returns the total games won count for Even and Odd. |

**Method Glossary for class Scoreboard:**

public **scoreboard**( ):

purpose: Instantiates a scoreboard object with 0 initial values for all counters.

public void **printScore**(scoreboard):

param1: Takes in a scoreboard object, representing the scoreboard object used to keep track of the score of the Game of Games.

purpose: Printing out all the stored values in the scoreboard object being passed to this method.

public int **getTotal**( ):  
return: integer, representing the count of the total games played.

purpose: Returning the count of total games played, and using it in conjunction with printScore() to print the scoreboard to the console.

public int **getTotalWins**( ):

return: integer, representing the count for total wins amongst all games.

purpose: Returning the count of total games won, and using it in conjunction with printScore() to print the scoreboard to the console.

public int **getTotalLoss**( ):

return: integer, representing the count for total losses amongst all games.  
purpose: Returning the count of total games lost, and using it in conjunction with printScore() to print the scoreboard to the console.

public int **getThread**( ):

return: integer, representing the count of total games played for Find the Red Thread.

purpose: Returning the count of total games played for Find the Red Thread, used in conjunction with printScore() to print the scoreboard to the console.

public int **getThreadWins**( ):

return: integer, representing the count of total games won for Find the Red Thread.

purpose: Returning the count of total games won for Find the Red Thread, used in conjunction with printScore() to print the scoreboard to the console.

public int **getThreadLoss**( ):

return: integer, representing the count of total games lost for Find the Red Thread.

purpose: Returning the count of total games lost for Find the Red Thread, used in conjunction with printScore() to print the scoreboard to the console.

public int **getThimble**( ):

return: integer, representing the count of total games played for Find the Thimble.

purpose: Returning the count of total games played for Find the Thimble, used in conjunction with printScore() to print the scoreboard to the console.

public int **getThimbleWins**( ):

return: integer, representing the count of total games won for Find the Thimble.

purpose: Returning the count of total games won for Find the Thimble, used in conjunction with printScore() to print the scoreboard to the console.

public int **getThimbleLoss**( ):

return: integer, representing the count of total games lost for Find the Thimble.

purpose: Returning the count of total games lost for Find the Thimble, used in conjunction with printScore() to print the scoreboard to the console.

public int **getGuess**( ):

return: integer, representing the count of total games played for Guess the Number.

purpose: Returning the count of total games played for Guess the Number, used in conjunction with printScore() to print the scoreboard to the console.

public int **getGuessWins**( ):

return: integer, representing the count of total games won for Guess the Number.

purpose: Returning the count of total games won for Find the Thimble, used in conjunction with printScore() to print the scoreboard to the console.

public int **getGuessLoss**( ):

return: integer, representing the count of total games lost for Guess the Number.

purpose: Returning the count of total games lost for Guess the Number, used in conjunction with printScore() to print the scoreboard to the console.

public int **getToss**( ):

return: integer, representing the count of total games played for Coin Toss.

purpose: Returning the count of total games played for Coin Toss, used in conjunction with printScore() to print the scoreboard to the console.

public int **getTossWins**( ):

return: integer, representing the count of total games won for Coin Toss.

purpose: Returning the count of total games won for Coin Toss, used in conjunction with printScore() to print the scoreboard to the console.

public int **getTossLoss**( ):

return: integer, representing the count of total games lost for Coin Toss.

purpose: Returning the count of total games lost for Coin Toss, used in conjunction with printScore() to print the scoreboard to the console.

public int **getEO**( ):

return: integer, representing the count of total games played for Even and Odd.

purpose: Returning the count of total games played for Even and Odd, used in conjunction with printScore() to print the scoreboard to the console.

public int **getEOWins**( ):

return: integer, representing the count of total games won for Even and Odd.

purpose: Returning the count of total games won for Even and Odd, used in conjunction with printScore() to print the scoreboard to the console.

public int **getEOLoss**( ):

return: integer, representing the count of total games lost for Even and Odd.

purpose: Returning the count of total games lost for Even and Odd, used in conjunction with printScore() to print the scoreboard to the console.

public void **setEO**(int):

param1: integer, used to specify the number by which to increment the total games played and the games played for Even and Odd.

purpose: Incrementing the game counters for total games played, and Even and Odd games.

public void **setEOWins**(int):

param1: integer, used to specify the number by which to increment the total games won for total games won counter and the Even and Odd counter.

purpose: Incrementing the game counters for total games won, and Even and Odd games.

public void **setEOLoss**(int):

param1: integer, used to specify the number by which to increment the total games lost for total games lost counter and the Even and Odd counter.

purpose: Incrementing the game counters for total games lost, and Even and Odd games.

public void **setGuess**(int):

param1: integer, used to specify the number by which to increment the total games played and the games played for Guess the Number.

purpose: Incrementing the game counters for total games played, and Guess the Number games.

public void **setGuessWins**(int):

param1: integer, used to specify the number by which to increment the total games won for total games won counter and the Guess the Number counter.

purpose: Incrementing the game counters for total games won, and Guess the Number games.

public void **setGuessLoss**(int):

param1: integer, used to specify the number by which to increment the total games lost for total games lost counter and the Guess the Number counter.

purpose: Incrementing the game counters for total games lost, and Guess the Number games.

public void **setToss**(int):

param1: integer, used to specify the number by which to increment the total games played and the games played for Coin Toss.

purpose: Incrementing the game counters for total games played, and Coin Toss games.

public void **setTossWins**(int):

param1: integer, used to specify the number by which to increment the total games won for total games won counter and the Coin Toss counter.

purpose: Incrementing the game counters for total games won, and Coin Toss games.

public void **setTossLoss**(int):

param1: integer, used to specify the number by which to increment the total games lost for total games lost counter and the Coin Toss counter.

purpose: Incrementing the game counters for total games lost, and Coin Toss games.

public void **setThread**(int):

param1: integer, used to specify the number by which to increment the total games played and the games played for Find the Red Thread.

purpose: Incrementing the game counters for total games played, and Find the Red Thread games.

public void **setThreadWins**(int):

param1: integer, used to specify the number by which to increment the total games won for total games won counter and the Find the Red Thread counter.

purpose: Incrementing the game counters for total games won, and Find the Red Thread games.

public void **setThreadLoss**(int):

param1: integer, used to specify the number by which to increment the total games lost for total games lost counter and the Find the Red Thread counter.

purpose: Incrementing the game counters for total games lost, and Find the Red Thread games.

public void **setThimble**(int):

param1: integer, used to specify the number by which to increment the total games played and the games played for Find theThimble.

purpose: Incrementing the game counters for total games played, and Find theThimble games.

public void **setThimbleWins**(int):

param1: integer, used to specify the number by which to increment the total games won for total games won counter and the Find theThimble counter.

purpose: Incrementing the game counters for total games won, and Find theThimble games.

public void **setThimbleLoss**(int):

param1: integer, used to specify the number by which to increment the total games lost for total games lost counter and the Find theThimble counter.

purpose: Incrementing the game counters for total games lost, and Find theThimble games.

| Game |
| --- |
| + won: integer + lost: integer + played: integer |
| + Game(): Instructor method. Creates new Game object with all fields set to 0. + getWon(): Accessor method to the won field. Returns an integer. + getLost(): Accessor method to the lost field. Returns an integer. + getPlayed(): Accessor method to the played field. Returns an integer.  + won(): Increments the won field by 1. Returns void. + lost(): Increments the lost field by 1. Returns void. + played(): Increments the played field by 1. Returns void. |

**Method Glossary for class Game:**

public **Game**( ):

purpose: Instantiates a Game object with default - 0’s for all the fields - values.

public int **getWon**( ):

return: integer, representing the number of games won.

purpose: Returns the number of games won, which is used to keep track of the overall score to display each time the user returns to the main menu.

public int **getLost**( ):

return : integer, representing the number of games lost.

purpose: Returns the number of games lost, which is used to keep track of the overall score to display each time the user returns to the main menu.

public int **getPlayed**( ):

return: integer, representing the number of games played.

purpose: Returns the number of games played, which is used to keep track of the overall score to display each time the user returns to the main menu.

public void **won**( ):

purpose: Increments the ‘won’ count, used whenever a game ends, this method would increment the won count for the instance of the game played. The main purpose of this method is updating the scoreboard.

public void **lost**( ):

purpose: Increments the ‘lost’ count, used whenever a game ends. This method would increment the lost counter for the instance of the game played. The main purpose of this method is updating the scoreboard.

public void **played**( ):

purpose: Increments the ‘played’ count, used whenever a game ends. This method would increment the played counter for the instance of the game played. The main purpose of this is to updating the scoreboard and to keep track of the numbers of games played.

| EvenOdd |
| --- |
| # rounds: integer # roundsPlayed: integer # roundsWon: integer # roundsLost: integer # troundsPlayed: integer # troundsWon: integer # troundsLost: integer # even: boolean |
| + EvenOdd( ): Instructor method. Creates an EvenOdd instance with all fields set to 0 and false.  + displayInstructions( ): Prints out the Even and Odd game instructions to the console. + setEvenOdd(): Prompts user input to change the value of the ‘even’ flag. + hmRounds( ): Prompts the user to select the number of rounds to play Even and Odd out of. + play( ): Main game loop, where the user gets to play Even and Odd against the CPU. + updateScore( ): Increments the total rounds played, total rounds won, and total rounds lost fields, and resets the rounds won, played, and lost fields.  + getTotalRounds( ): Returns the number of total rounds played. + getTotalWon( ): Returns the number of total rounds won. + getTotalLost( ): Returns the number of total rounds lost. + compTurn(int, int): Generates and returns a random number between the inclusive range of the two integers being passed to the method. |

**Method Glossary for class EvenOdd:**

public **EvenOdd**( ):

purpose: Creates an instance of the class EvenOdd, extending the class Game.   
  
public void **displayInstructions**( ):

purpose: Displays the Even & Odd game instructions for the user to read.

public void **setEvenOdd**( ):  
purpose Prompts the user to input a value between 1 and 2 to set the even flag in order for the computer to calculate a winner for the game.

public void **hmRounds**( ):

purpose: Prompts the user to input an odd integer value specifying the number of rounds they want to play best of.

public void **play**( ):

purpose: Prompts the user to take their turn in the Even and Odd game. It does so until the user has enough rounds won, or lost, to determine a winner for the game.

public void **updateScore**( ):

purpose: Increment the total rounds played, won, and lost fields by the number of rounds played after play() is done executing, and then reset the rounds played, won, and lost in case the user wants to play another game before returning to the main menu.

public int **getTotalRounds**( ):

return: integer, representing the number of total rounds played in this instance of Even and Odd.  
purpose: Updating the scoreboard.

public int **getTotalWon**( ):

return: integer, representing the number of total rounds won in this instance of Even and Odd.  
purpose: Updating the scoreboard.

public int **getTotalLost**( ):

return: integer, representing the number of total rounds lost in this instance of Even and Odd.  
purpose: Updating the scoreboard.

public int **compTurn**(int, int):

param1: integer, takes in a minimum value for the range of the computer turn.  
param2: integer, takes in a maximum value for the range of the computer turn.   
return: a random integer value in the range [param1, param2].

purpose: generates the computer turn for the user to engage with in play().

|  |
| --- |
| **Thread** |
| + spools: integer + userTurn: boolean  + winnerTurn: boolean  + Red: String  + max: integer |
| + Thread(): Constructor method. Instantiates elements of Thread class.  + displayInstructions(): Auxiliary method. Displays instructions to play the game.  + spoolsNum(): Auxiliary method. Prompts the user for a number of threads to be entered before a turn ends.  + turn(): Auxiliary method. Facilitates switching turns between the user and the CPU.  + winner(): Auxiliary method. Switches a flag to signal that the red thread has been found.  + reset(): Auxiliary method. Resets the flags of the class Thread to their default values.  + play(): Main method of the Find the Red Thread game. |

**Method Glossary for class Thread:**

public int **getVal**():

return: Returns the field value inputVal.

purpose: instantiates the field inputVal.

public **Thread**():

purpose: Constructor class to instantiate fields of the Thread class.

Public void **displayInstructions**():

purpose: Displays instructions to play the game.

public void **spoolsNum()**:

purpose: Prompts the user for the number of threads to be entered before a turn ends.

public void **turn**():

purpose: Facilitates switching turns between the user and the CPU.

public void **winner**(): Switches a flag to signal that the red thread has been found.

public void **reset**():

purpose: Resets the flags of the class Thread to their default values.

public void play():

purpose: Main method of the Find the Red Thread game.

|  |
| --- |
| **coinToss** |
| + userWins: integer + compWins: integer + totalRounds: integer  + HEADS: integer  + TAILS: integer  + user: integer |
| + displayInstructions(): Helper method for the coinToss class. It displays instructions on how to play the game. + randomGen(int, int): Helper method. Generates a random number between 1 and 2. Param1: integer specifies the minimum range and Param2: integer specifies maximum range. + hmRounds(): Accessor method. Allows a user to choose a number of rounds to play out of for an instance of the coin toss game. + headsOrTails(): Accessor method. Allows user to pick heads or tails.  + play(): Built-in class. Allows user to interact with coinToss game. |

**Method Glossary for class coinToss:**

public **displayInstructions**():

purpose: Displays instructions on how to play the game.

public int **randomGen**(int, int):

return: Returns an integer - the randomly generated number.

purpose: Generates a random number between 1 and 2

public void **hmRounds**():

purpose: Allows a user to choose a number of rounds to play out of for an instance of the coinToss game. The number of rounds the user is allowed to pick has to be an odd integer.

public void **headsOrTails**():

purpose: Allows user to pick heads or tails.

public void **play**( ):

purpose: Helper class to drive the coinToss game.

|  |
| --- |
| **getInput** |
| + inputVal: integer |
| + getInput(): Instructor method, Sets the inputVal variable to its default.  + getVal(): Auxiliary method. Returns the field value inputVal + gamePick(): Auxiliary method. Prompts the user for a value between 1 and 6, inclusive to pick what game they want to play. + startOrReturn(): Auxiliary method. Prompts the user for a value between 1 and 2, inclusive. It is used in the driver method for the different games startup.  + playAgain(): Auxiliary method. Prompts the user for a value between 1 and 2, inclusive. Allows user deciding to return to the main menu or play the game again.  + bestOf(): Auxiliary method. Lets the user decide how many rounds they want to play out of after they choose to play certain games.  + evenOddPick(): Auxiliary method. Prompts the user to input an integer between 1 and 2, inclusive. Lets the user pick their choice of even or odd for the “Even or Odd Game”.  + evenOddTurn(): Auxiliary method. Prompts the user to enter an integer value between 1 and 5, inclusive. User input acts as their turn in a round of Even and Odd.  + spools(): Auxiliary method. Finds the Red Thread in the “Find the Red Thread Game”.  + enterRangeMin(): Auxiliary method. Computes guess the number and returns the minimum number for the “Guess the Number Game”.  + enterRangeMax(int): Auxiliary method. Returns the maximum number. Param1: integer specifies the minimum number returned from the enterRangeMax() auxiliary method in the “Guess the Number Game”.  + numGuess(int, int): Auxiliary method. Performs the number guess for the computer. Param1: integer specifies the minimum number, Param2: specifies the maximum number.  + Usernumber(int, int): Auxiliary method. Performs the number guess for the user. Param1: integer specifies the minimum number. Param2: integer specifies the maximum number.  + headsOrTailsPick(): Auxiliary method. Calls input for heads or tails in the “Coin Toss Game”. |

**Method Glossary for class getInput:**

public **displayInstructions**():

purpose: Displays instructions on how to play the game.

public int **randomGen**(int, int):

return: Returns an integer - the randomly generated number.

purpose: Generates a random number between 1 and 2

Public void **getInput**():

purpose: Sets the inputVal variable to its default.

public int **getVal**():

return: Returns the field value inputVal.

purpose: instantiates the field inputVal.

public void **gamePick**():

purpose: Allows user to pick what game they want to play.

Public void **startOrReturn**():

purpose: Used across all games to allow a user to start a game or exit to the main menu.

public void **playAgain**():

purpose: Used across all games to allow a user to start a game or exit to the main menu.

public void **bestOf**():

purpose: Allows the user to decide how many rounds they want to play out of after they choose to play certain games.

public void **evenOddPick**():

purpose: Allows the user to pick their choice of even or odd for the “Even or Odd Game”.

public void **evenOddTurn**():

purpose: Allows the user to enter their turn or move for even or odd.

public void **spools**():

purpose: Finds the Red Thread in the “Find the Red Thread Game”.

public int **enterRangeMin**():

return: Returns an integer, the minimum input value for “Guess the Number Game”

purpose: Computes guess the number and returns the minimum number for the “Guess the Number Game”.

public int **enterRangeMax**(int):

return: Returns an integer, the maximum value for “Guess the Number Game”

purpose: Returns the maximum number. Param: integer specifies the maximum number returned from the enterRangeMax() auxiliary method in the “Guess the Number Game”.

Public int **numGuess**(int, int):

return: Returns the number of guesses for the input value.

purpose: Performs the number of guesses for the computer. Param1: integer specifies the minimum number, Param2: specifies the maximum number.

public int **Usernumber(int, int)**:

return:Returns an integer, the user pick.

purpose: Performs the number guess for the user. Param1: integer specifies the minimum number. Param2: integer specifies the maximum number.

public void **headsOrTailsPick()**:

purpose: Calls input for heads or tails in the “Coin Toss Game”.

| **GuesstheNumber** |
| --- |
| + ChanceTaken: Integer |
| + GuesstheNumber(): Instructor method.  + ComputerGenerate():Computer randomly generates number, returns an integer.  + displayInstruction(): Instruction display, returns void + UserGuess(): Returns an integer.  + GuessCompare(): Compare with number return from UserGuess and ComputerGenerate. Returns void. |

**Method Glossary for class Scoreboard:**

public **GuesstheNumber():**

Purpose: Instantiates a GuesstheNumber object with no parameters.

public int **ComputerGenerate(int MinRange, int MaxRange):**

Purpose: Get the number that randomly generated by computer

Parameters: 1. Integer MinRange which is the lower bound of guessing number range input.

2. Integer MaxRange which is the upper bound of guessing number range input.

Returns: Integer, representing the number that Computer picks (answer of the game).

public int **displayInstruction():**

Purpose: Display all the instruction to users.

Parameters: None

Return: void

public int **UserGuess(int min, int max):**

Purpose: Get user’s input of the number they guess.

Parameters: 1. Integer Min which is the lower bound of guessing number range input.

2. Integer Max which is the upper bound of guessing number range input.

Returns: Integer, representing the number user picks.

public void **GuessCompare():**

Purpose: This function is for the actual game process.

Parameters: None

Return: void

| **Spool** |
| --- |
| + String[] colors: String array  + max: Integer  + COLORS: Integer  + RED:Integer  + Spool[] box: Spool type array  + color:String |
| + Spool(): Instructor method, creates new spool object  + spoolBox(): Auxiliary method automates the creation of unique thread, returns void.  + picked(): Take in integer, return a spool object  + BoxPick(): Returns an integer, returns a integer.  + getColor(): Return a string. |

**Method Glossary for class Spool:**

**p****ublic Spool():**

Purpose: Default Constructor

**public Spool(int):**

Purpose: Instructor method for the class Spool. It assigns each new Spool object with a color based on the integer value being passed to it.

Params: Integer

**public void spoolBox():**

Purpose: Simulates a box of threads by utilizing an array data structure. This is an auxiliary

method which automates the creation of unique threads in an organized fashion.

Params: N/A

Returns: void.

**public Spool Picked(int):**

Purpose: simulates picking a thread out of an array of threads.

Params: An integer

Returns:a Spool object.

**public int boxPick(int):**

Purpose: generates a random number between 0 and the integer value being passed to it.

Params: An integer

Returns: An integer, The value being returned is within the range [0,max].

**public String getColor():**

Purpose: Accessor method for the field color of a Spool instance.

Params: N/A.

Returns: A string which contains the color associated with a spool object.

| **PlayGame** |
| --- |
|  |
| * printMenu() : Static method prints out the main menu to the console. * main(String): Static method, driver method for the game of games. |

**Method Glossary for class PlayGames:**

public static void **printMenu**( ):

Purpose: Display game options for the user.

public static **main(**String):

param1: Takes in arguments if there are any. Not meant to be used.

Purpose: Driver method for the game of games.

| **Thimble** |
| --- |
| * rounds: integer * roundsWon: integer * roundsLost: integer * LEFT: integer * RIGHT: integer |
| * Thimble(): Instructor method. * displayInstructions() Displays instructions to the console. * randomGen(int, int): Generates a number within the range of the two values being passed to the method. * hmRounds(): Prompts user to pick how many rounds to play best out of. * play(): Main Find The Thimble game method. |

**Method Glossary for class Thimble():**

public void **hmRounds**( ):

purpose: Prompts the user to input an odd integer value specifying the number of rounds they want to play best of.

public **Thimble():**

purpose: Instantiate a Thimble game object.

public void **displayInstructions**( ):

purpose: Displays the Even & Odd game instructions for the user to read.

public int **randomGen(**int, int):

param1: Integer value, specifies the minimum range of values.

param2: Integer value, specifies the maximum range of values.

return: Integer, a random value between param1 and param2.

Purpose: Generate a random value, and using that value to assign the thimble to a random hand.

public void **play():**

purpose: lets the user play the Find the Thimble game.