

Forum:[Hack Forums](#)

Thread: [Mini-Project #5 \(LiN 01\)](#)

Karil Votek - July 4, 2014, 10:22 am

Been busy with other things lately and had little time to code, but working back on putting effort into learning. Noticed [Psifour](#) created a thread that contained a project list from another source. Plan to work on that list from 0-99.

[package GameFiles]

HeadsTails class:

Spoiler ([Click to View](#))

PHP Code

```
<?php
package gamefiles;
/*Author: Karil Votek
 * Purpose: Class simulates a game of heads and tails.
 * Summary: Heads or tails is chosen->player guesses heads or tails->program tells if they are correct or not
 */
import java.util.Arrays;
import java.util.List;
import java.util.Random;
import java.util.Scanner;

public class HeadsTails {

    /*Private variables:
     * tryAgainLoop - Boolean used to repeat loop in tryAgain() method-see tryAgain()
     * wins/losses - Amount of times players guesses correctly or not
     * options - List that holds the two possible options for the game [heads/tails]
     * coinToss - 'Randomly' selects one of the possible from options[]
     * playerGuess - Variable to hold player's guess->see passGuess()
     * input - Scanner to receive input from player
     */

    boolean tryAgainLoop;
    private int wins, losses;
    private List options = Arrays.asList("heads", "tails");
    private Random coinToss = new Random();
    private String playerGuess, toss;
    private Scanner input = new Scanner(System.in);

    //Method used to run the methods below (can see as gear that gets other gears moving)
    public void run() {
        //Obtaining the guess from player->see passGuess method
        playerGuess = passGuess();

        //Running method to choose random string from options[] then determining whether they were correct
```

```

ct
    flip();
}

/*Flip method:
*'Randomly' sets toss to heads or tails
*Action taken after determining whether toss is equal to the player's guess or not
*/
private void flip() {
    //Setting toss to heads or tails
    toss = (String) options.get(coinToss.nextInt(options.size()));
    //Player guessed correctly
    if (toss.equals(playerGuess)) {
        System.out.println("You are correct!\n");
        wins++; //wins increases by one
    } //Player guessed incorrectly
    else if (!toss.equals(playerGuess) && validateInput() == true) {
        System.out.println("Sorry, you lost...\n");
        losses++; //losses increases by one
    } else {
        System.out.println("Please enter heads or tails.\n");
    }
    //Runs tryAgain method->See tryAgain() for more details
    tryAgain();
}

//Method used to see if player would like to play again
private void tryAgain() {
    //Setting tryAgainLoop to false to run the loop below
    tryAgainLoop = false;
    //Loop to ensure player does not proceed if they enter anything other than yes or no
    while(!tryAgainLoop) {
        //Holds input from player->See passChoice for more details
        String choice = passChoice();
        //If player wishes to play the game again, the run method is called.->See run() above
        if (choice.equals("yes")) {
            run();
            tryAgainLoop = true;
        } //If the player quits/enters anything other than "yes", their wins/losses are given.
        else if (choice.equals("no")) {
            String results = String.format("Wins: %,d | Losses: %,d\n", wins,
                losses);
            System.out.println(results + "\nThank you for playing!\n");
            //exitGame (Games static boolean) is set to true to break out of game loop
            Games.exitGame = true;
            tryAgainLoop = true;
        }
    }
}

//String method for checking if player would like to play again
private String passChoice() {
    System.out.println("Would you like to play again? (yes/no): ");
    //Returning their text input from the console
    return input.nextLine().toLowerCase(); //Added toLowerCase() in case player uses caps
}

```

```

    }
    //String method for asking player to choose Heads or Tails
    private String passGuess() {
        System.out.print("Heads or Tails: ");
        //Returning their text input from the console.
        return input.nextLine().toLowerCase(); //Added toLowerCase() in case player uses caps
    }
    //Method to check whether player entered something other than heads or tails
    private boolean validateInput() {
        return options.contains(playerGuess);
    }
}

```

HigherLower class:

Spoiler ([Click to View](#))

PHP Code

```

<?php
package gamefiles;
/*Author: Karil Votek
 * Purpose: Class simulates a number guessing game.
 * Summary: Random number is selected->user guesses what number is->program tells if they are c
orrect/higher/or lower
 */
import java.util.Random;
import java.util.Scanner;

public class HigherLower {

    /*Private variables:
     * randomNumber - Holds the random number generated by the java.util.Random class
     * playerGuess - Used to store the value the player enters->See passGuess()
     * attempts - Keeps track of amount of times player has guessed for correct number
     * rand - Instance of Random class that is used to 'randomly' choose a number 1-100
     * input - Instance of Scanner class that is used to receive user input for game
     * started - Boolean used to signal player has guessed once->See passGuess()
     * isHigher - Boolean used when deciding which message player receives->See run() and passGu
    ess()
    */

    private int randomNumber, playerGuess, attempts;
    private Random rand = new Random();
    private Scanner input = new Scanner(System.in);
    private boolean started = false;
    private boolean isHigher;
    String playAgain, winningMessage;

    //Method used to run the methods below (can see as gear that gets other gears moving)
    public void run() {
        //Making randomNumber a 'random' integer between 0 and 100
        randomNumber = rand.nextInt(100);
        //Loop used to keep player guessing until they are correct
        while (playerGuess != randomNumber) {

```

```

//Setting playerGuess variable to the user's input->See passGuess()
playerGuess = passGuess();

//If the player's guess is higher than randomNumber, isHigher becomes false->See passGuess()
if (playerGuess > randomNumber) {
    isHigher = false;

//Otherwise, isHigher is set to true (playerGuess is lower than randomNumber)->See passGuess()
} else {
    isHigher = true;
}
//Increasing attempts by one because player tried to guess
attempts++;
}
//If player guesses correctly, loop exits to here.
winningMessage = String.format(
    "You have won in %,d attempt(s)!\n", attempts);
System.out.println(winningMessage);
//Running tryAgain()->See tryAgain()
tryAgain();
}
//Method used to find out if player wishes to play again
private void tryAgain() {
    //Setting playAgain to input from user
    playAgain = passChoice();
    //If player enters yes, reset attempt to 0, started to false, and starts run() to replay game
    if (playAgain.equals("yes")) {
        attempts = 0;
        started = false;
        run();
        //Otherwise, game exits loop in Game class
    } else {
        System.out.println("Thank you for playing!\n");
        //exitGame (Games static boolean) is set to true to break out of game loop
        Games.exitGame = true;
    }
}
//String method used for passing player's input on playing again
private String passChoice() {
    System.out.println("Would you like to play again? (yes/no): ");
    return input.nextLine().toLowerCase(); //Added toLowerCase() in case player uses caps
}
//String method used to pass player's guess
private int passGuess() {
    //If this is player's first guess, sends this message to let them know number range.
    if (started == false) {
        System.out.println("Guess a number between 0-100: ");
        //Setting started variable to true to skip this message next time
        started = true;

//If this is not player's first guess and previous guess was lower than randomNumber, this gets called
    } else if (started == true && isHigher == true) {

```

```

        System.out.println("Guess a higher number: ");

//If this is not player's second guess and previous guess was higher than randomNumber, this gets called
    } else if (started == true && isHigher == false) {
        System.out.println("Guess a lower number: ");
    }
    //Returning player's input->see tryParse()
    return tryParse(input.nextLine());
}
//Int method used to validate the player's input
private int tryParse(String string) {
    //Creating a variable to pass value back
    int value;
    //Attempting to parse the player's input in a try/catch statement
    try {
        //Setting value to the input
        value = Integer.parseInt(string);
        //If player inserts value over 100, value is set to previous guess
        if(value > 100) {
            value = playerGuess;
            System.out.println("Cannot guess higher than 100. \n");
        }
        //If input is lower than 0, value is set to previous guess
        } else if(value < 0) {
            value = playerGuess;
            System.out.println("Cannot guess a number lower than 0. \n");
        }
    } catch (NumberFormatException e) {
        //Value is set to previous guess if input cannot be parsed
        value = playerGuess;
        System.out.println("Please enter a number between 0-100 only");
    }
    //Returning the player's input/previous guess if mistakes were made
    return value;
}
}
}

```

Games class:

Spoiler ([Click to View](#))

PHP Code

```
<?php
```

```
package gamefiles;
```

```
/*Author: Karil Votek
```

```
* Purpose: Class is the main menu for the program.
```

```
* Summary: Presents user with game choices->game loop for their choice is ran/option to end program->
```

```
*/
```

```
import java.util.Scanner;
```

```
public class Games {
```

```
    /*Private variables:
```

- * playerChoice - variable to hold player's input of 1, 2, or 3->see displayOptions()
- * gameOne - Instance of HeadsTails class->see HeadsTails.java
- * gameTwo - Instance of HigherLower class->see HigherLower();
- * input - Instance of Scanner class used for receiving player input
- * exitGame - Boolean used to exit out of gameOne or gameTwo loops

```

*/

private int playerChoice;
private HeadsTails gameOne = new HeadsTails();
private HigherLower gameTwo = new HigherLower();
private Scanner input = new Scanner(System.in);
static boolean exitGame;
//Method used to run the methods below (can see as gear that gets other gears moving)
public void run() {
    loop();
    System.out.println("Your game session has ended.");
}
//Main game loop that holds loops for gameOne and gameTwo
private void loop() {
    //As long as the player's game choice does not equal 3 (means quit), loop runs
    while(playerChoice != 3) {
        //Setting exitGame to false to allow HeadsTails and HigherLower loop to be ran again
        exitGame = false;
        //Displaying game choice menu for player->see displayOption()
        displayOptions();
        //Setting playerChoice to player's input->see choice()
        playerChoice = choice();
        //If player's choice is lower than 1 or greater than 3, informs them to choose 1, 2, or 3
        if(playerChoice < 1 || playerChoice > 3) {
            System.out.println("Please enter a 1, 2, or 3 for the choices above.");
        } else {
            while(exitGame == false && playerChoice == 1) {
                //HeadsTails game runs if player entered 1 for the choice
                gameOne.run();
            }
            while(exitGame == false && playerChoice == 2) {
                //HigherLower runs if player entered 2 for the choice
                gameTwo.run();
            }
        }
    }
}
//Method used to display options player can choose from a start of program/exit of one of games
private void displayOptions() {
    System.out.println("Welcome to the menu! \n"
        + "Please choose an option below: \n"
        + "1 - HeadsTails - Try to guess whether it is heads or tails! \n"
        + "2 - HigherLower - Try to guess the random number between 0-100! \n"
        + "3 - Quit - When you are finish playing, enter 3. \n"
        + "Choice: ");
}
//Int method used to return the player's choice->see tryParse()

```

```

private int choice() {
    return tryParse(input.nextLine());
}
//Int method used to validate the player's input
private int tryParse(String string) {
    //Creating a variable to pass value back
    int value;
    //Attempting to parse the player's input in a try/catch statement
    try {
        //Setting value to the input
        value = Integer.parseInt(string);
    } catch (NumberFormatException e) {
        //If player's input cannot be parsed, value is set to 4 to inform player to enter 1, 2, or 3
        value = 4;
    }
    //Returning the player's input/set value if mistake was made
    return value;
}
}

```

[package votek]

Main class:

Spoiler ([Click to View](#))

PHP Code

```
<?php
```

```
package votek;
```

```
import gamefiles.Games;
```

```
public class Main {
```

```

    public static void main(String[] args) {
        Games games = new Games();
        games.run();
    }
}

```

If you know any way I can shape up the code above or tips/advice, please let me know.