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

/**
 * Rock Paper Scissors Game - Blueprint Class
 *
 * @author Jasur Shukurov
 * @version Version 3.1
 * @version 12/12/2018
 */


public class Player {


    // Initializing variables
    private static int wins = 0;
    private static int losses = 0;
    private static int ties = 0;

    private static boolean stop = false;

    private static Scanner scn = new Scanner(System.in);

    /*
     * determineWinner - method which finds who won the game
     *
     * @param p1 - player1 choise
     * @param p2 - player2 choice
     *
     */

    public static String determineWinner(String p1, String p2) {
        if (p1.equalsIgnoreCase(p2)) {
            ties();
            return "Draw!";
        } else if ((p1.equalsIgnoreCase("rock") && p2.equalsIgnoreCase("scissors"))
                || (p1.equalsIgnoreCase("paper") && p2.equalsIgnoreCase("rock"))
                || (p1.equalsIgnoreCase("scissors") && p2.equalsIgnoreCase("paper"))) {
            wins();
            return "Congratulation Player 1 won!";
        } else {
            losses();
            return "Oh, sorry Player 2 won!";
        }

    }

    /*
     * ties - method which adds one to ties Statistics
     */
    private static void ties() {
        ties++;

    }

    /*
     * losses - method which adds one to losses Statistics
     */
    private static void losses() {
        losses++;

    }

    /*
     * wins - method which adds one to wins Statistics
     */
    private static void wins() {
        wins++;

    }

    /*
     * printStats() - this method generates game statistics
     *
     * @return String, returns game statistics
     *
     */
    public static String printStats() {

        return "\nYour Statistics:\nTies: " + ties + "\nWins: " + wins + "\nLosses: " + losses;

    }

    /*
     * readUserChoice() - this method reads user input
     *
     * @return String, returns user input
     *
     */
    public static String generateComputerChoice() {
        Random rnd = new Random();
        int player2 = rnd.nextInt(3) + 1;

        if (player2 == 1) {
            return "rock";
        } else if (player2 == 2) {
            return "paper";
        } else {
            return "scissors";
        }

    }

    /*
     * readUserChoice() - this method reads user input
     *
     * @return String, returns user input
     *
     */
    private static String readUserChoice() {

        System.out.println("Please type rock, paper or scissors: ");
        String player1 = scn.nextLine();

        while (!valid(player1)) {
            System.out.println("Invalid input, please try again: ");
            player1 = scn.nextLine();
        }

        return player1;

    }

    /*
     * valid() - this method checks if user input is valid or not.
     *
     * @return boolean, if user input is valid it returns true, otherwise it
     * returns false
     *
     */
    private static boolean valid(String player1) {

        switch (player1.toLowerCase()) {
        case "rock":
            return true;
        case "scissors":
            return true;
        case "paper":
            return true;
        default:
            return false;
        }

    }

    /*
     * playAgain() - method which asks from user, does s(he) want to play again
     * ?
     *
     */
    private static void playAgain() {
        System.out.println("Do you want to play again? ");
        String userInput = scn.nextLine();
        if (userInput.equalsIgnoreCase("No")) {
            stop = true;
        }
    }

    /*
```

```java
     * choosingType() - method which asks user to choose what type of game (s)he
     * wants to play. It can be Human agains Human, Computer vs Human or
     * Computer against Computer.
     *
     * @return int, it returns user choice
     */
    public static int choosingType() {

        int type = scn.nextInt();
        scn.nextLine();
        while (type < 1 || type > 3) {
            System.out.println("You typed invalid value, please try again!");
            type = scn.nextInt();
            scn.nextLine();
        }

        return type;

    }

    /*
     * start() - is method which starts game
     */
    public String start() {

        int type = choosingType();

        if (type == 1)
            return playerVsComputer();// player agains computer
        else if (type == 2)
            return playerVsPlayer();// player agains player
        else
            return computerVsComputer();// computer against computer

    }

    /*
     * computerVsComputer() - computer plays against computer
     */
    private static String computerVsComputer() {

        while (stop == false) {
            playComputerAgainstComputer();
            playAgain();
        }

        return printStats();
    }

    /*
     * playComputerAgainstComputer()
     *
     * @return String
     */
    public static void playComputerAgainstComputer() {
        String player1 = generateComputerChoice();
        String player2 = generateComputerChoice();

        System.out.println("Computer 1 chose - " + player1);
        System.out.println("Computer 2 chose - " + player2);

        String winner = determineWinner(player1, player2);

        System.out.println(winner);
    }

    /*
     * playerVsPlayer() - method to play against another player
     */
    private static String playerVsPlayer() {

        while (stop == false) {
            playAgainstPlayer();
            playAgain();
        }
        return printStats();
    }

    /*
     * playAgainstPlayer()
     *
     * @return String
     *
     */
    public static void playAgainstPlayer() {
        String player1 = readUserChoice();
        String player2 = readUserChoice();
        String winner = determineWinner(player1, player2);

        System.out.println(winner);
    }

    /*
     * playerVsComputer() - method to play against computer
     *
     */
    private static String playerVsComputer() {
        while (stop == false) {
            playAgainstComputer();
            playAgain();
        }
        return printStats();
    }

    /*
     * playAgainstComputer()
     *
     * @return String
     *
     */
    public static void playAgainstComputer() {
        String player1 = readUserChoice();
        String player2 = generateComputerChoice();
        System.out.println("Computer chose - " + player2);
        String winner = determineWinner(player1, player2);

        System.out.println(winner);
    }

    /*
     * printInstruction() - returns instrution
     *
     * @return String
     */
    public String printInstruction() {
        String instruction = "Instruction!!!\nPlease type \n1 if you want play against Computer\n2 "
                + "to play against another Player \n3 to Computer play against Computer!\n";
        return instruction;

    }

}

```