```
1 import java.util.Random;
 2 import java.util.Scanner;
 5 * Rock Paper Scissors Game - Blueprint Class
6 *
7 * @author Jasur Shukurov
 8 * @version Version 3.1
 9 * @version 12/12/2018
10 */
11
12
13 public class Player {
15
      // Initializing variables
16
17
      private static int wins = 0;
18
      private static int losses = 0;
      private static int ties = 0;
19
20
21
      private static boolean stop = false;
22
23
      private static Scanner scn = new Scanner(System.in);
24
25
       * determineWinner - method which finds who won the game
26
27
28
       * @param p1 - player1 choise
        * @param p2 - player2 choice
29
30
31
32
33
      public static String determineWinner(String p1, String p2) {
34
           if (p1.equalsIgnoreCase(p2)) {
               ties();
return "Draw!";
35
36
37
           } else if ((p1.equalsIgnoreCase("rock") && p2.equalsIgnoreCase("scissors"))
                    II (p1.equalsIgnoreCase("paper") && p2.equalsIgnoreCase("rock"))
II (p1.equalsIgnoreCase("scissors") && p2.equalsIgnoreCase("paper"))) {
38
39
40
               wins();
return "Congratulation Player 1 won!";
41
42
           } else {
43
               losses();
               return "Oh, sorry Player 2 won!";
44
45
           }
46
47
      }
48
49
       * ties - method which adds one to ties Statistics
50
51
      private static void ties() {
53
           ties++;
54
55
      }
56
       * losses - method which adds one to losses Statistics
58
59
60
      private static void losses() {
61
           losses++;
63
      }
64
65
       * wins - method which adds one to wins Statistics
66
67
68
      private static void wins() {
69
           wins++;
70
71
      }
73
        * printStats() - this method generates game statistics
74
75
76
       * @return String, returns game statistics
77
78
79
      public static String printStats() {
```

```
return "\nYour Statistics:\nTies: " + ties + "\nWins: " + wins + "\nLosses: " + losses;
 81
 82
 83
       }
 84
 85
          readUserChoice() - this method reads user input
 86
 87
 88
          @return String, returns user input
 89
 90
 91
       public static String generateComputerChoice() {
 92
           Random rnd = new Random();
 93
           int player2 = rnd.nextInt(3) + 1;
 94
           if (player2 == 1) {
    return "rock";
 95
 96
             else if (player2 == 2) {
 97
 98
               return "paper";
 99
           } else {
100
               return "scissors";
101
102
103
       }
104
105
        * readUserChoice() - this method reads user input
106
107
108
          @return String, returns user input
109
110
       private static String readUserChoice() {
111
112
113
           System.out.println("Please type rock, paper or scissors: ");
114
           String player1 = scn.nextLine();
115
116
           while (!valid(player1)) {
117
               System.out.println("Invalid input, please try again: ");
118
               player1 = scn.nextLine();
119
           }
120
           return player1;
121
122
123
       }
124
125
        126
127
        * @return boolean, if user input is valid it returns true, otherwise it
128
          returns false
129
130
131
132
       private static boolean valid(String player1) {
133
           switch (player1.toLowerCase()) {
134
135
           case "rock":
136
               return true;
137
           case "scissors'
138
               return true:
139
           case "paper":
140
               return true;
141
           default:
142
               return false;
143
144
145
       }
146
147
148
          playAgain() - method which asks from user, does s(he) want to play again
149
150
151
152
       private static void playAgain() {
153
           System.out.println("Do you want to play again? ");
154
           String userInput = scn.nextLine();
           if (userInput.equalsIgnoreCase("No")) {
155
156
               stop = true;
157
158
       }
159
       /*
160
```

```
161
         * 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
162
163
          Computer against Computer.
164
165
          @return int, it returns user choice
166
167
       public static int choosingType() {
168
169
           int type = scn.nextInt();
170
            scn.nextLine();
171
            while (type < 1 || type > 3) {
                System.out.println("You typed invalid value, please try again!");
172
173
                type = scn.nextInt();
               scn.nextLine();
174
175
           }
176
177
            return type;
178
179
       }
180
181
182
          start() - is method which starts game
183
184
       public String start() {
185
           int type = choosingType();
186
187
188
            if (type == 1)
189
               return playerVsComputer();// player agains computer
190
            else if (type == 2)
               return playerVsPlayer();// player agains player
191
192
193
               return computerVsComputer();// computer against computer
194
195
       }
196
197
198
        * computerVsComputer() - computer plays against computer
199
200
       private static String computerVsComputer() {
201
202
            while (stop == false) {
203
               playComputerAgainstComputer();
204
               playAgain();
205
           }
206
207
            return printStats();
208
       }
209
210
          playComputerAgainstComputer()
211
212
213
          @return String
214
215
       public static void playComputerAgainstComputer() {
216
            String player1 = generateComputerChoice();
217
            String player2 = generateComputerChoice();
218
            System.out.println("Computer 1 chose - " + player1);
219
220
            System.out.println("Computer 2 chose - " + player2);
221
222
            String winner = determineWinner(player1, player2);
223
224
            System.out.println(winner);
225
       }
226
227
228
          playerVsPlayer() - method to play against another player
229
       private static String playerVsPlayer() {
230
231
232
            while (stop == false) {
               playAgainstPlayer();
233
234
               playAgain();
235
236
            return printStats();
237
       }
238
239
        * playAgainstPlayer()
240
```

```
241
                             * @return String
242
243
244
                         public static void playAgainstPlayer() {
245
                                       String player1 = readUserChoice();
246
247
                                       String player2 = readUserChoice();
248
                                       String winner = determineWinner(player1, player2);
249
250
                                       System.out.println(winner);
251
                        }
252
253
254
                            * playerVsComputer() - method to play against computer
255
256
                         private static String playerVsComputer() {
257
258
                                       while (stop == false) {
259
                                                    playAgainstComputer();
260
                                                     playAgain();
261
262
                                        return printStats();
263
                        }
264
265
                           * playAgainstComputer()
266
267
268
                                   @return String
269
270
271
                         public static void playAgainstComputer() {
272
                                       String player1 = readUserChoice();
273
                                        String player2 = generateComputerChoice();
                                      System. out.println("Computer chose - " + player2);
String winner = determineWinner(player1, player2);
274
275
276
277
                                       System.out.println(winner);
278
                        }
279
280
                                  printInstruction() - returns instrution
281
282
283
                             * @return String
284
285
                         public String printInstruction() {
                                       \textbf{String instruction} = \texttt{"Instruction!!!} \\ \texttt{nPlease type } \\ \texttt{n1 if you want play against Computer} \\ \texttt{n2 "} \\ \texttt{n2 instruction!!!} \\ \texttt{n3 instruction!!!} \\ \texttt{n3 instruction!!!} \\ \texttt{n4 instruction!!} \\ \texttt{n4 instruction!} \\ \texttt{n4 instruc
286
287
                                                                   + "to play against another Player \n3 to Computer play against Computer!\n";
288
                                       return instruction;
289
290
                        }
291
292 }
```