

```
1 import java.util.Random;
2
3 public class Dice {
4     private int value;
5
6     public Dice() {
7         this.value = 0;
8     }
9
10    public void roll() {
11        Random random = new Random();
12
13        this.value = random.nextInt(6) + 1;
14    }
15
16    public int getValue() {
17        return this.value;
18    }
19 }
20
```

```
1 import java.util.Scanner;
2
3 public class Game {
4
5     private Entity player;
6     private Entity computer;
7     private int    gamesNumber;
8
9     public Game() {
10         this.player = new Entity();
11         this.computer = new Entity();
12         this.gamesNumber = 0;
13     }
14
15     public void displayMenu() {
16         Scanner scanner = new Scanner(System.in);
17         int choice = 0;
18
19         System.out.println("1 - Play the dice war game");
20         System.out.println("2 - Display statistics");
21         System.out.println("3 - Exit the game");
22         String line = scanner.next();
23
24         try {
25             choice = Integer.parseInt(line);
26         } catch (Exception e) {
27             e.printStackTrace();
28             System.out.println("Sorry, your request has not
29 been understood.");
30             this.displayMenu();
31         }
32
33         switch (choice) {
34             case 1:
35                 this.play();
36                 break;
37             case 2:
38                 System.out.println("Number of played games
39 : " + this.gamesNumber);
40                 System.out.println("The player won " + this
41 .player.getWonGames() + " game(s).");
42                 System.out.println("The AI won " + this.
43 computer.getWonGames() + " game(s).");
44                 this.displayMenu();
45                 break;
46             case 3:
47                 return;
48             default:
49                 System.out.println("Sorry, your request has
50 not been understood.");
51         }
52     }
53 }
```

```

46         this.displayMenu();
47         break;
48     }
49 }
50
51 public void init() {
52     Scanner scanner = new Scanner(System.in);
53
54     System.out.println("Do you want to play ? [yes/no]"
55 );
56     String line = scanner.nextLine();
57
58     if (line.toLowerCase().equals("yes")) {
59         this.play();
60     } else if (line.toLowerCase().equals("no")) {
61         System.out.println("Goodbye !");
62         this.displayMenu();
63         return;
64     } else {
65         System.out.println("Sorry, your request has not
66 been understood.");
67         this.init();
68     }
69 }
70
71 public void play() {
72     int playerValue = 0;
73     int computerValue = 0;
74     this.player.rollDice();
75     this.computer.rollDice();
76
77     System.out.print("Player has drawn : (");
78     for (int i = 0 ; i < this.player.getDices().length;
79 i++) {
80         playerValue += this.player.getDices()[i].
81         getValue();
82         System.out.print(this.player.getDices()[i].
83         getValue());
84         if (i + 1 < this.player.getDices().length) {
85             System.out.print(", ");
86         }
87     }
88     System.out.println(") = " + playerValue);
89
90     System.out.print("Computer has drawn : (");
91     for (int i = 0 ; i < this.computer.getDices().
92     length; i++) {
93         computerValue += this.computer.getDices()[i].
94         getValue();

```

```
89         System.out.print(this.computer.getDices()[i].
    getValue());
90         if (i + 1 < this.computer.getDices().length) {
91             System.out.print(", ");
92         }
93     }
94     System.out.println(" = " + computerValue);
95
96     if (playerValue < computerValue) {
97         this.computer.gameWon();
98         System.out.println("Computer WON.");
99     } else if (playerValue > computerValue) {
100         this.player.gameWon();
101         System.out.println("Player WON.");
102     } else {
103         System.out.println("This is a TIE.");
104     }
105
106     this.gamesNumber += 1;
107
108     this.init();
109 }
110
111 }
112
```

```
1
2 public class Main {
3
4     public static void main(String[] args) {
5         Game game = new Game();
6
7         game.displayMenu();
8     }
9 }
10
```

```
1 public class Entity {
2
3     private Dice[]    dices;
4     private int       gamesWon;
5
6     public Entity() {
7         this.dices = new Dice[2];
8         this.dices[0] = new Dice();
9         this.dices[1] = new Dice();
10        this.gamesWon = 0;
11    }
12
13    public void rollDice() {
14        for (Dice dice: this.dices) {
15            dice.roll();
16        }
17    }
18
19    public Dice[] getDices() {
20        return this.dices;
21    }
22
23    public int getWonGames() {
24        return this.gamesWon;
25    }
26
27    public void gameWon() {
28        this.gamesWon += 1;
29    }
30
31 }
32
```

```
1 - Play the dice war game
2 - Display statistics
3 - Exit the game
1
Player has drawn : (1, 4) = 5
Computer has drawn : (2, 3) = 5
This is a TIE.
Do you want to play ? [yes/no]
no
Goodbye !
1 - Play the dice war game
2 - Display statistics
3 - Exit the game
2
Number of played games : 1
The player won 0 game(s).
The AI won 0 game(s).
1 - Play the dice war game
2 - Display statistics
3 - Exit the game
3
Process finished with exit code 0
```