

Table of contents

1- Code	2 - 10
2- Program execution	11 - 12

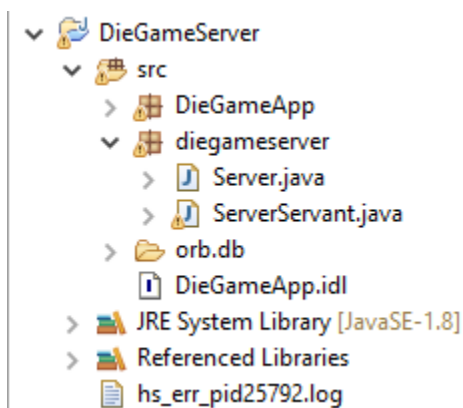
1- Code: (Explanation in code comments)

DieGameApp.idl:

```
module DieGameApp {  
  
    //Player  
    interface PlayerInt {  
        void callback (in string msg);  
    };  
  
    //Server  
    interface ServerInt {  
        long MRandom (in string player_name);  
        void RegisterCallback (in PlayerInt player);  
    };  
  
};
```

SERVER (Explanation in code Comments)

DieGameServer folder



Server.java

```
1 package diegameserver;
2
3 import org.omg.CORBA.ORB;
4 import org.omg.CosNaming.NameComponent;
5 import org.omg.CosNaming.NamingContextExt;
6 import org.omg.CosNaming.NamingContextExtHelper;
7 import org.omg.PortableServer.POA;
8 import org.omg.PortableServer.POAHelper;
9
10 import DieGameApp.*;
11
12 public class Server {
13
14     public static void main (String args []){
15
16         try {
17
18             //Init orb
19             ORB orb = ORB.init(args, null);
20
21             //get rootpoa ref
22             POA rootpoa = POAHelper.narrow(orb.resolve_initial_references("RootPOA"));
23             rootpoa.the_POAManager().activate();
24
25             // create servant register it with ORB and get reference as "server"
26             ServerServant serverservant = new ServerServant ();
27             serverservant.setORB(orb);
28             org.omg.CORBA.Object objRef = rootpoa.servant_to_reference(serverservant);
29             ServerInt server = ServerIntHelper.narrow(objRef);
30
31             //get namingcontextext
32             NamingContextExt ncRef = NamingContextExtHelper.narrow(orb.resolve_initial_references("NameService"));
33
34             //bind servant reference in naming context in naming context
35             String name = "Server";
36             NameComponent path [] = ncRef.to_name(name);
37             ncRef.rebind(path, server);
38
39             System.out.println("Server listening");
40
41             //Wait for players invokations
42             orb.run();
43
44         } catch (Exception e){
45             e.printStackTrace();
46         }
47
48         System.out.println("Server exit");
49     }
50 }
51
52 }
```

ServerServant.java

```
1 package diegameserver;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.SQLException;
6 import java.sql.Statement;
7 import java.util.ArrayList;
8 import java.util.Random;
9
10 import org.omg.CORBA.ORB;
11
12 import DieGameApp.*;
13
14 // Hold player's information (name, score, player_ref : PlayerServant)
15 class Player_Info {
16     public String name ;
17     public int score ;
18     public PlayerInt player_ref; // Reference to the Player Servant passed with the RegisterCallback function
19
20     @Override
21     public String toString () {
22         return name + " " + score + " " + player_ref;
23     }
24 }
25
26 public class ServerServant extends ServerIntPOA {
27
28     private static int counter = 0 ; // When counter % 7 == 0, reset all the variables in this class
29
30     private String player_name ;
31     private int player_score ;
32     private PlayerInt player_ref;
33
34     private Connection conn = null ;
35
36     // Will hold the 7 player's info that register -- Clear when counter % 7 == 0
37     private ArrayList <Player_Info> players_info_list = new ArrayList <Player_Info> () ;
38
39     private static ORB orb;
40     public void setORB (ORB _orb){
41         orb = _orb;
42     }
43
44     private Connection getConnection() throws Exception{
45         String driver="com.microsoft.sqlserver.jdbc.SQLServerDriver";
46         String url="jdbc:sqlserver://localhost:1433;databaseName=CORBA_DIE_GAME_DB";
47         String username="sa";
48         String password="";
49         Class.forName(driver);
50         return DriverManager.getConnection(url,username,password);
51     }
52 }
```

```

53 private void InsertPlayer (String player_name, int score){ // Insert player in database
54
55     String query = "INSERT INTO scores (player_name, score) VALUES "
56         + "("+player_name+", '"+score+"'";
57
58     try {
59         conn = getConnection () ;
60         Statement stmt = conn.createStatement () ;
61
62         stmt.execute(query);
63
64     }catch (Exception e){
65         e.printStackTrace();
66         System.exit(-1);
67     }finally {
68         if (conn != null){
69             try {
70                 conn.close();
71             } catch (SQLException e) {
72                 e.printStackTrace();
73             }
74         }
75     }
76
77 }
78
79 // Return a random number and set player_name and player_score
80 public int MRandom(String _player_name) {
81     Random random = new Random (System.currentTimeMillis()) ;
82     int random_number = random.nextInt(7) + 1;
83
84     player_name = _player_name ;
85     player_score = random_number ;
86
87     return random_number ;
88 }
89
90
91 // Get the PlayerServant reference to pass it as a parameter to the callback function of the player
92 // add a Player_Info to the players_info_list
93 // Insert the player's information in the database
94 // increment counter by 1
95 // Check for a winner
96 public void RegisterCallback(PlayerInt playerref) {
97     player_ref = playerref ; // Get Player Servant reference
98
99     Player_Info player_info = new Player_Info () ;
100     player_info.name = player_name ;
101     player_info.score = player_score;
102     player_info.player_ref = player_ref;
103
104     players_info_list.add (player_info); //add a Player_Info to the players_info_list
105
106     InsertPlayer (player_name, player_score); // Insert in database
107
108     counter ++ ;
109
110     CheckWinner () ;
111 }
112
113

```

```

114 private void CheckWinner () {
115
116     if (counter % 7 != 0){ // If number of current players is not 7, don't check for a winner (return)
117         return ;
118     }
119
120     //Get the player with the highest score from the players_info_list
121     Player_Info winner_info = this.players_info_list.get(0) ;
122     for ( int i = 1 ; i < this.players_info_list.size() ; i ++ ){
123         Player_Info current_player_info = this.players_info_list.get(i);
124         if (current_player_info.score > winner_info.score){
125             winner_info = current_player_info;
126         }
127     }
128
129
130     String results_msg = "Player: " + winner_info.name + " is the winner with a score of: " + winner_info.score;
131
132     String players_left_msg = "Players left: "; // List of the players that left the game before doing the callback
133
134     for ( int i = 0 ; i < this.players_info_list.size() ; i ++ ){
135
136         // If an exception is thrown, that means this player is not a valid object: this player left the game
137         try {
138             this.players_info_list.get(i).player_ref.callback(results_msg);
139             this.players_info_list.get(i).player_ref.callback(players_left_msg);
140         }catch (Exception e){
141             players_left_msg += this.players_info_list.get(i).name;
142         }
143
144     }
145
146
147     //Reset variables
148     counter = 0;
149     this.players_info_list.clear();
150
151 }
152
153 }

```

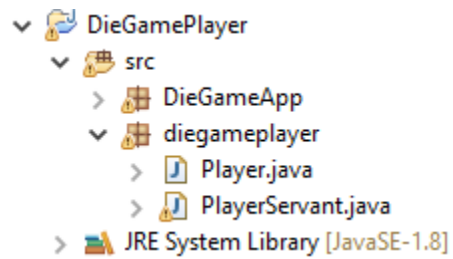
DATABASE

SQL

```
create table scores (  
    id int identity (1,1),  
    player_name varchar (50),  
    score int  
);
```

PLAYER

DieGamePlayer folder



Player.java

```
1 package diegameplayer;
2
3 import java.io.BufferedReader;
4
5
6
7 // Called after a timer exceeds it time - Interrupt the thread to end orb.run ()
8 class PlayerExit extends TimerTask {
9
10     Thread player_thread;
11
12     public PlayerExit (Thread _player_thread){
13         player_thread = _player_thread;
14     }
15
16     // execute when the timer exceeds
17     public void run () {
18         player_thread.interrupt();
19     }
20 }
21
22 public class Player extends Thread {
23
24     private static ServerInt server ;
25
26     public static void main (String args []) throws IOException{
27
28         String player_name;
29
30         System.out.println("Enter your name:");
31         System.out.flush();
32
33         BufferedReader inl=new BufferedReader(new InputStreamReader(System.in));
34         player_name = inl.readLine();
35
36         try {
37
38             //Init ORB
39             ORB orb = ORB.init(args, null);
40
41             //get root POA and ref
42             POA rootpoa = POAHelper.narrow(orb.resolve_initial_references("RootPOA"));
43             rootpoa.the_POAManager().activate();
44
45             //create player servant register it with ORB and get reference as "player"
46             PlayerServant playerservant = new PlayerServant () ;
47             playerservant.setORB(orb);
48             playerservant.setName(player_name);
49             org.omg.CORBA.Object objRef = rootpoa.servant_to_reference(playerservant);
50             PlayerInt player = PlayerIntHelper.narrow(objRef);
51
52             //Get NamingContextExt
53             NamingContextExt ncRef = NamingContextExtHelper.narrow(orb.resolve_initial_references("NameService"));
54
55         }
56     }
57 }
```



```

64
65 //Get Server servant from naming context
66 String name = "Server";
67 server = ServerIntHelper.narrow(ncRef.resolve_str(name));
68
69 //START REGISTERING CLIENT WITH SERVER
70
71 //get random number
72 int random_number = server.MRandom(player_name);
73
74 //print out random number
75 System.out.println("Your number is " + random_number);
76
77 //register with server to be called back
78 server.RegisterCallback(player);
79
80
81 // Interrupt thread after 5 min to remove the block of orb.run ()
82 Timer t = new Timer () ;
83 t.schedule(new PlayerExit (Thread.currentThread()), 5 * 60 * 1000); // Call PlayerExit().run() after 5 minutes
84
85 orb.run();
86
87 }catch (Exception e){
88 // If the exception is caused by an interrupt --> interrupt the thread
89 if (e.getClass().getName().equalsIgnoreCase("java.lang.ArithmeticException ")){
90 Thread.currentThread().interrupt();
91 }
92
93 e.printStackTrace(System.out);
94 }
95
96
97 //Player left
98 System.out.println("Left");
99
100 }
101
102 }

```

PlayerServant.java

```
1 package diegameplayer;
2
3 import org.omg.CORBA.ORB;
4
5
6 public class PlayerServant extends PlayerIntPOA {
7
8     private String name ;
9
10
11     private static ORB orb;
12     public void setORB (ORB _orb) {
13         orb = _orb ;
14     }
15
16     //Used in Player.java to set the name of the player
17     public void setName (String name){
18         this.name = name ;
19     }
20
21     //called by the server
22     public void callback(String msg) {
23         System.out.println("From server: " + msg);
24     }
25
26 }
```

2- Program execution:

ORBD + SERVER:

```

C:\> Select Command Prompt - java diegameserver.Server -ORBInitialPort 1050 -ORBInitialHost localhost

D:\>start orbd -ORBInitialPort 1050

D:\>cd Programs\Eclipse Mars\Java Workspace

D:\Programs\Eclipse Mars\Java Workspace>cd DieGameServer/src
To use the JDBC driver when compiling
D:\Programs\Eclipse Mars\Java Workspace\DieGameServer\src>set classpath=D:\mssql-jdbc-8.2.2.jre8.jar;

D:\Programs\Eclipse Mars\Java Workspace\DieGameServer\src>javac diegameserver/*.java

D:\Programs\Eclipse Mars\Java Workspace\DieGameServer\src>java diegameserver.Server -ORBInitialPort 1050 -ORBInitialHost localhost
Server listening

```

PLAYERS:

A screenshot of a Windows Command Prompt window titled "Command Prompt". The background is black with white text. The command prompt shows the following sequence of commands and outputs:

1. Initial directory: D:\Programs\Eclipse Mars\Java Workspace>
2. Change directory: cd DieGamePlayer/src
3. Compile: javac diegameplayer/*.java
4. Run (7 times): start java diegameplayer.Player -ORBInitialPort 1050 -ORBInitialHost localhost

The last seven run commands are enclosed in a red rectangular box. To the right of this box, the text "To create 7 players" is written in red. The window has standard Windows title bar controls (minimize, maximize, close) at the top right.

OUTPUT:

```
C:\Program Files (x86)\Common Files\Oracle\Java\javapath\java...
Enter your name:
player1
Your number is 3
From server: Player: player6 is the winner with a score of: 7
From server: Players left:
Left
```

```
C:\Program Files (x86)\Common Files\Oracle\Java\javapath\jav...
Enter your name:
player5
Your number is 1
From server: Player: player6 is the winner with a score of: 7
From server: Players left:
Left
```

```
C:\Program Files (x86)\Common Files\Oracle\Java\javapath\jav...
Enter your name:
player2
Your number is 4
From server: Player: player6 is the winner with a score of: 7
From server: Players left:
Left
```

```
C:\Program Files (x86)\Common Files\Oracle\Java\javapath\java.exe
Enter your name:
player7
Your number is 2
From server: Player: player6 is the winner with a score of: 7
From server: Players left:
Left
```

```
Select C:\Program Files (x86)\Common Files\Oracle\Java\javap...
Enter your name:
player3
Your number is 4
From server: Player: player6 is the winner with a score of: 7
From server: Players left:
Left
```

```
C:\Program Files (x86)\Common Files\Oracle\Java\javapath\java...
Enter your name:
player6
Your number is 7
From server: Player: player6 is the winner with a score of: 7
From server: Players left:
Left
```

```
C:\Program Files (x86)\Common Files\Oracle\Java\javapath\java.e...
Enter your name:
player4
Your number is 4
From server: Player: player6 is the winner with a score of: 7
From server: Players left:
Left
```

DATABASE:

id	player_name	score
6	player1	3
7	player2	4
8	player3	4
9	player4	4
10	player5	1
11	player6	7
12	player7	2