K. Abhishek
TCOA76

DBMSL

Assignment No. : 12

Aim

Write a program to implement MongoDB database connectivity with PHP/Python/Java. Implement database navigation operations (add, delete, edit, etc) using ODBC/JDBC.

Objective

To study mongoDB connectivity with other programming languages.

Theory

MongoDB is a cross-platform, document oriented db that provides high performance, high availability and easy scalability. MongoDB works on concept of collection and document.
Before you start connecting MongoDB in your IDE, you need to make sure you have the mongoDB java drivers in the executable .jar format. If not, download it add it to your class path.

To connect db you need to specify the db name. If db doesn't exist, mongoDB automatically creates it.

Create a collection

To create a collection, createCollection() method of com.mongodb.client.mongodatabase class is used. You just pass the name of collection to method.

Getting/Selecting a collection

To get/select a collection from the db, getCollection() method of com.mongodb.client.mongodatabase is used. Pass the collection name to method.

Insert a document

To insert a doc into mongoDB, insert() method of com.mongodb.client.mongocollection class is used.

Retrive all documents

To select all doc from collection, find() method of com.mongodb.client.mongocollection class is used. This method returns a cursor, so you need to iterate this.

Update document

To update a doc from collection, updateOne() method of mongocollection class is used.

Delete a document

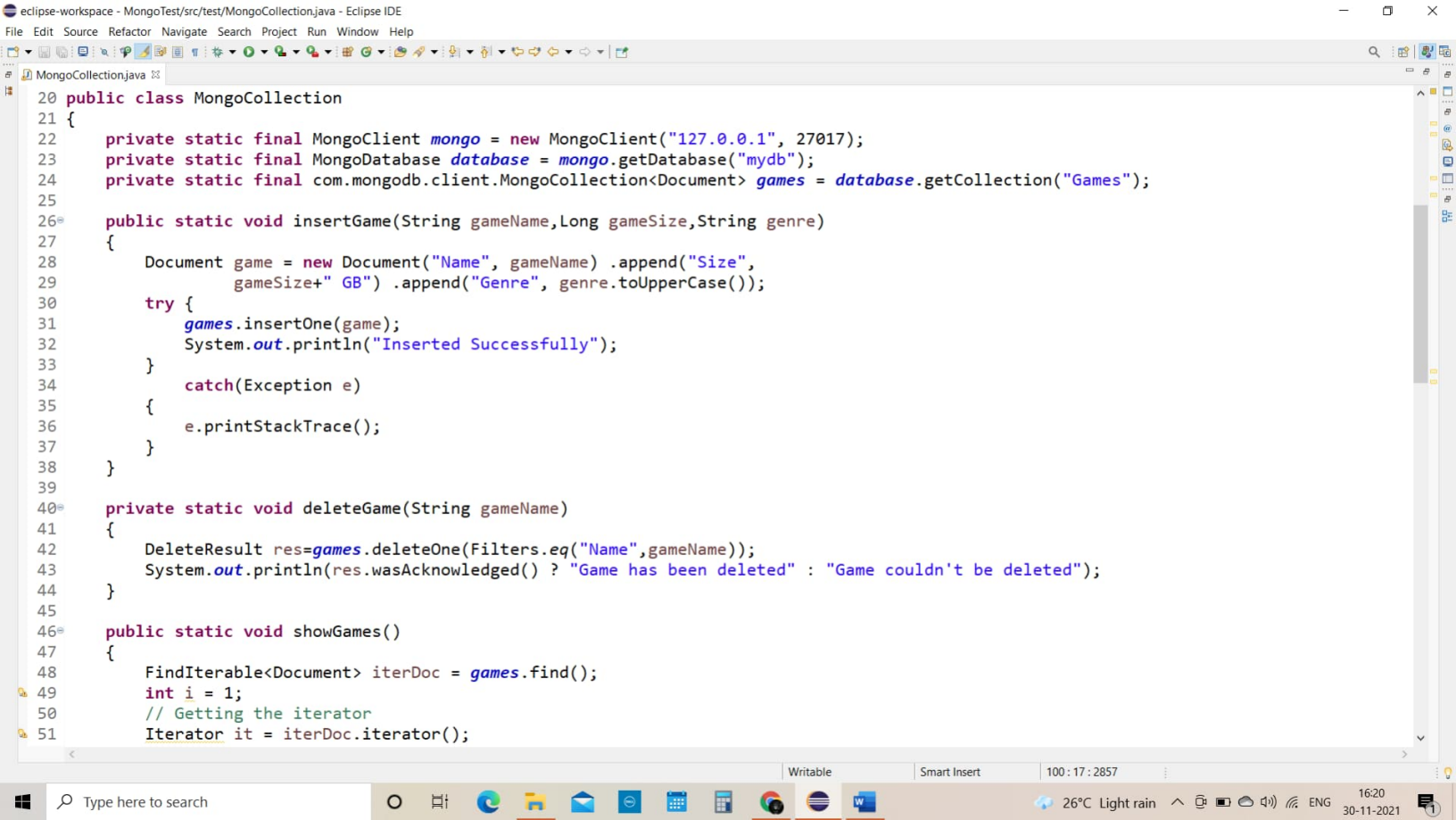
To delete a doc from collection, deleteOne() method of mongoCollection class is used.

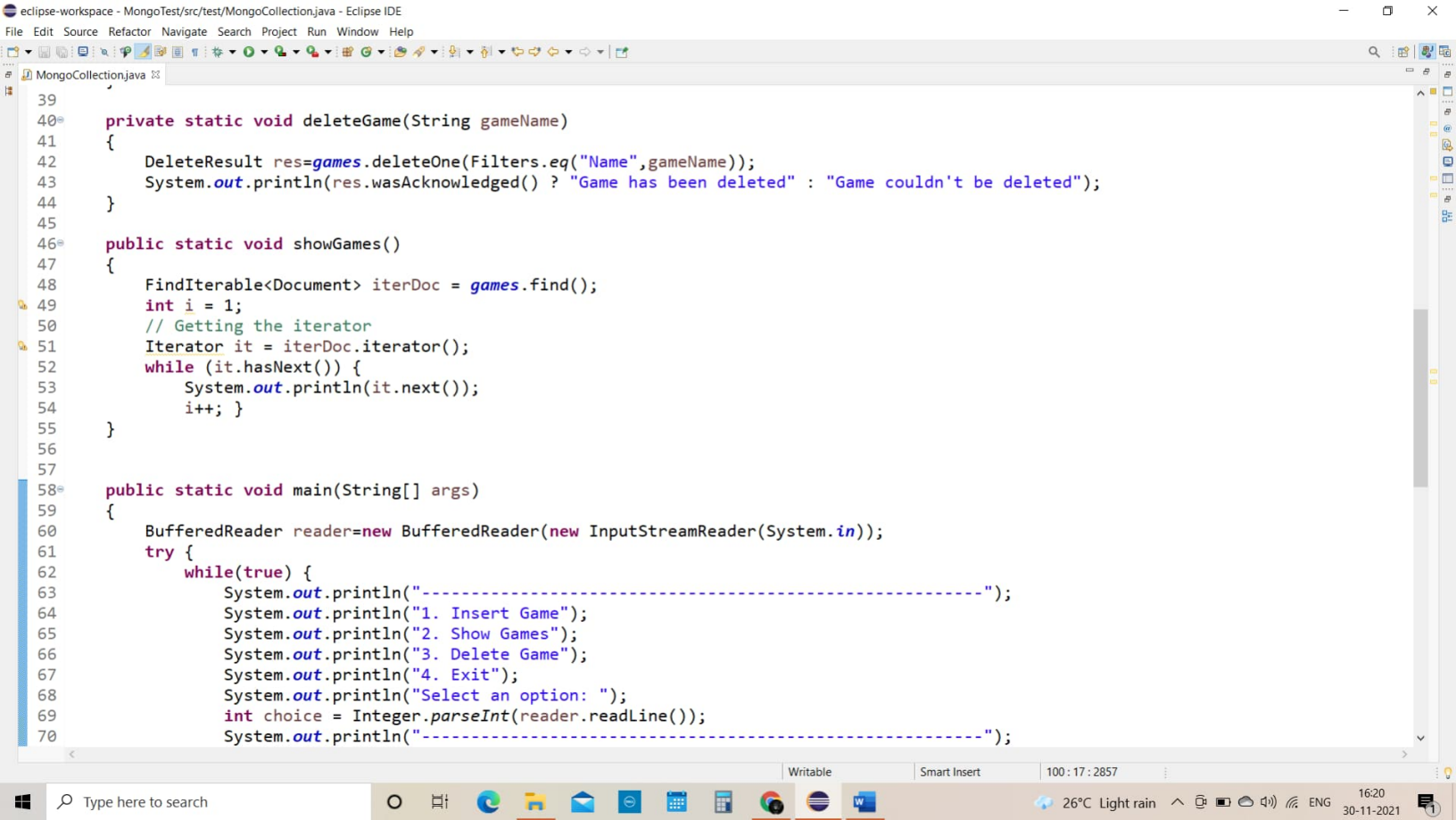
Conclusion

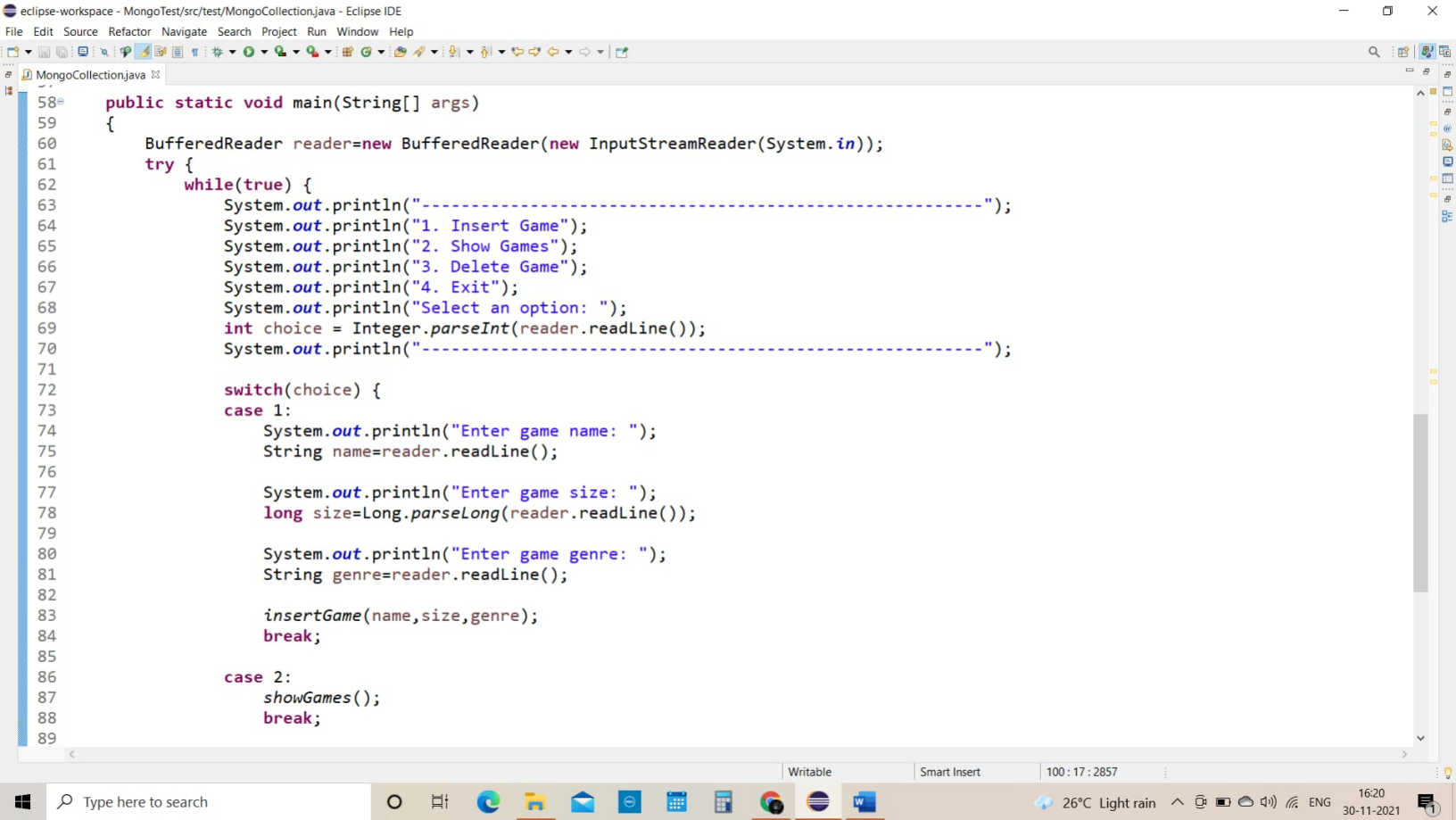
Thus, we have learnt the connectivity of JDBC with MongoDB.

MongoCollection.java

```
1  package test;
2
3  import java.io.BufferedReader;
4  import java.io.InputStreamReader;
5  import java.util.ArrayList;
6  import java.util.Iterator;
7  import java.util.List;
8
9  import org.bson.Document;
10
11 import com.mongodb.MongoClient;
12 import com.mongodb.client.FindIterable;
13 import com.mongodb.client.MongoDatabase;
14 import com.mongodb.client.model.Filters;
15 import com.mongodb.client.model.Updates;
16 import com.mongodb.client.result.DeleteResult;
17 import com.mongodb.client.result.*;
18
19
20 public class MongoCollection
21 {
22     private static final MongoClient mongo = new MongoClient("127.0.0.1", 27017);
23     private static final MongoDatabase database = mongo.getDatabase("mydb");
24     private static final com.mongodb.client.MongoCollection<Document> games = database.getCollection("Games");
25
26     public static void insertGame(String gameName,Long gameSize,String genre)
27     {
28         Document game = new Document("Name", gameName) .append("Size",
29                 gameSize+" GB") .append("Genre", genre.toUpperCase());
30         try {
31             games.insertOne(game);
32             System.out.println("Inserted Successfully");
```


eclipse-workspace - MongoTest/src/test/MongoCollection.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
MongoCollection.java
20 public class MongoCollection
21 {
22 private static final MongoClient mongo = new MongoClient("127.0.0.1", 27017);
23 private static final MongoDatabase database = mongo.getDatabase("mydb");
24 private static final com.mongodb.client.MongoCollection<Document> games = database.getCollection("Games");
25
26 public static void insertGame(String gameName,Long gameSize,String genre)
27 {
28 Document game = new Document("Name", gameName) .append("Size",
29 gameSize+" GB") .append("Genre", genre.toUpperCase());
30 try {
31 games.insertOne(game);
32 System.out.println("Inserted Successfully");
33 }
34 catch(Exception e)
35 {
36 e.printStackTrace();
37 }
38 }
39
40 private static void deleteGame(String gameName)
41 {
42 DeleteResult res=games.deleteOne(Filters.eq("Name",gameName));
43 System.out.println(res.wasAcknowledged() ? "Game has been deleted" : "Game couldn't be deleted");
44 }
45
46 public static void showGames()
47 {
48 FindIterable<Document> iterDoc = games.find();
49 int i = 1;
50 // Getting the iterator
51 Iterator it = iterDoc.iterator();
Writable
Smart Insert
100 : 17 : 2857
Type here to search
26°C Light rain
ENG
16:20
30-11-2021


eclipse-workspace - MongoTest/src/test/MongoCollection.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
MongoCollection.java
39
40 private static void deleteGame(String gameName)
41 {
42 DeleteResult res=games.deleteOne(Filters.eq("Name",gameName));
43 System.out.println(res.wasAcknowledged() ? "Game has been deleted" : "Game couldn't be deleted");
44 }
45
46 public static void showGames()
47 {
48 FindIterable<Document> iterDoc = games.find();
49 int i = 1;
50 // Getting the iterator
51 Iterator it = iterDoc.iterator();
52 while (it.hasNext()) {
53 System.out.println(it.next());
54 i++; }
55 }
56
57
58 public static void main(String[] args)
59 {
60 BufferedReader reader=new BufferedReader(new InputStreamReader(System.in));
61 try {
62 while(true) {
63 System.out.println("--");
64 System.out.println("1. Insert Game");
65 System.out.println("2. Show Games");
66 System.out.println("3. Delete Game");
67 System.out.println("4. Exit");
68 System.out.println("Select an option: ");
69 int choice = Integer.parseInt(reader.readLine());
70 System.out.println("--");
Writable
Smart Insert
100 : 17 : 2857
Type here to search
26°C Light rain
ENG
16:20
30-11-2021

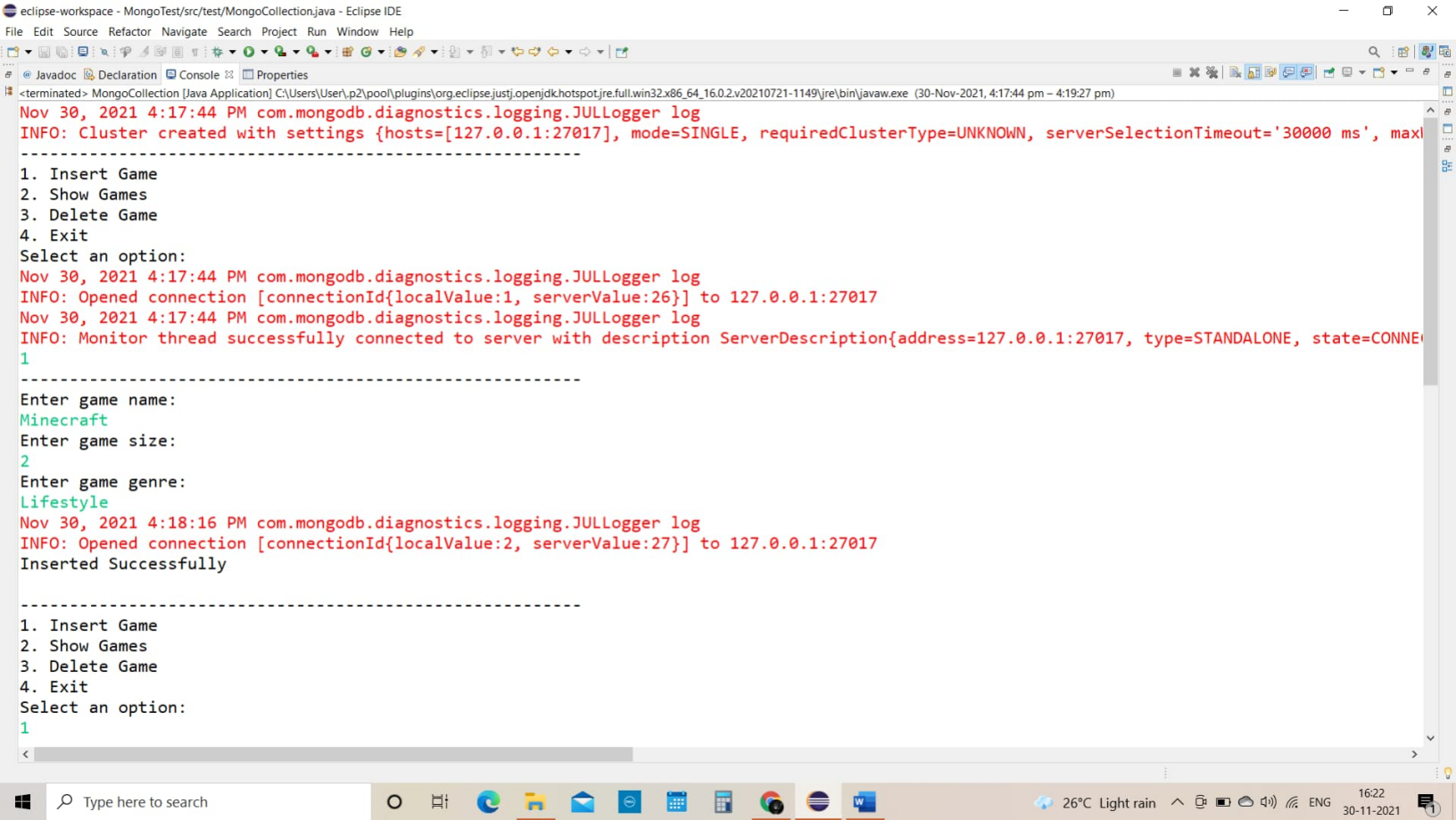

eclipse-workspace - MongoTest/src/test/MongoCollection.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
MongoCollection.java
public static void main(String[] args)
{
BufferedReader reader=new BufferedReader(new InputStreamReader(System.in));
try {
while(true) {
System.out.println("--");
System.out.println("1. Insert Game");
System.out.println("2. Show Games");
System.out.println("3. Delete Game");
System.out.println("4. Exit");
System.out.println("Select an option: ");
int choice = Integer.parseInt(reader.readLine());
System.out.println("--");
switch(choice) {
case 1:
System.out.println("Enter game name: ");
String name=reader.readLine();
System.out.println("Enter game size: ");
long size=Long.parseLong(reader.readLine());
System.out.println("Enter game genre: ");
String genre=reader.readLine();
insertGame(name,size,genre);
break;
case 2:
showGames();
break;
Writable
Smart Insert
100 : 17 : 2857
Type here to search
26°C Light rain
ENG
16:20
30-11-2021

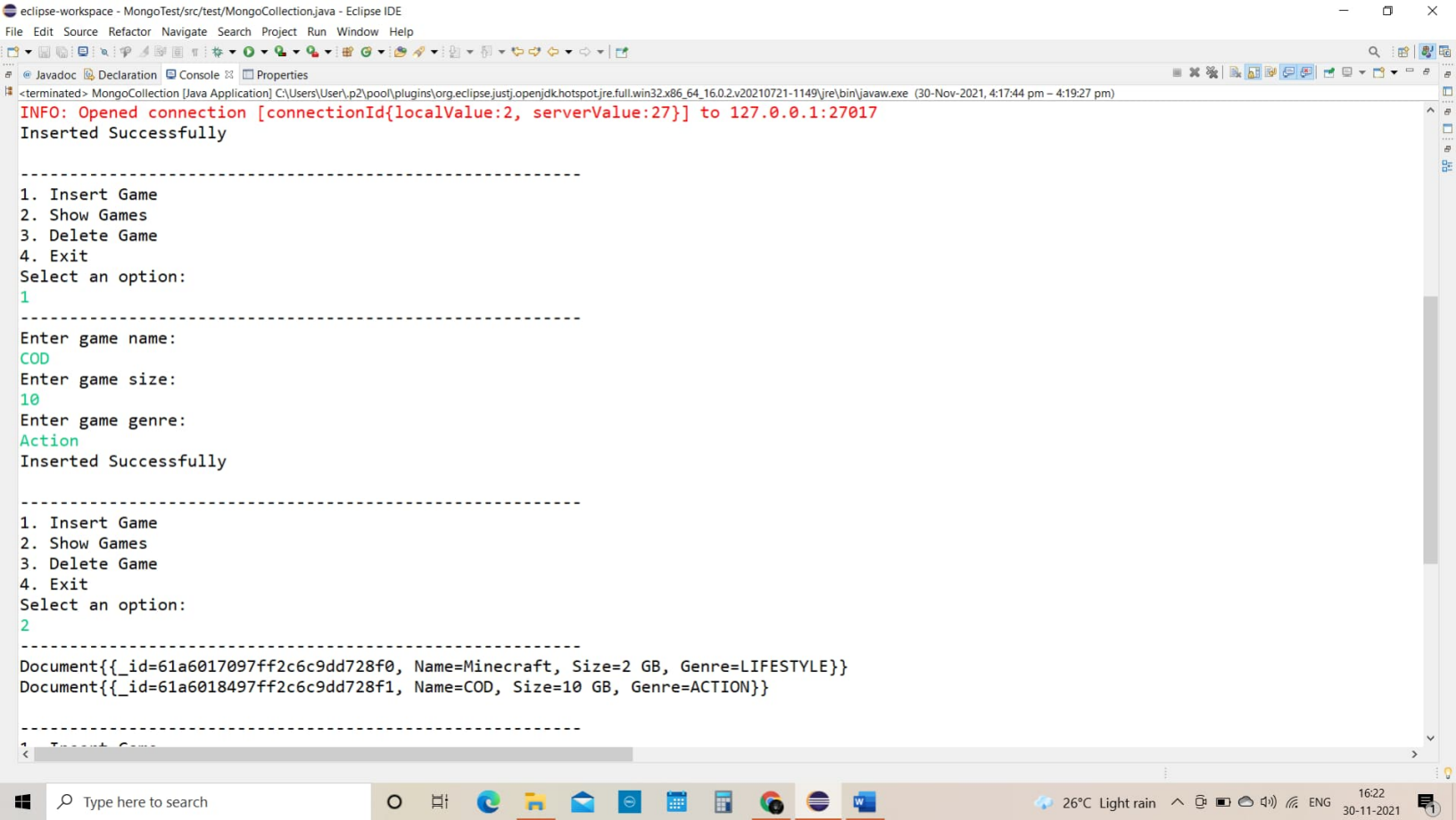
MongoCollection.java

```
72              switch(choice) {
73              case 1:
74                  System.out.println("Enter game name: ");
75                  String name=reader.readLine();
76
77                  System.out.println("Enter game size: ");
78                  long size=Long.parseLong(reader.readLine());
79
80                  System.out.println("Enter game genre: ");
81                  String genre=reader.readLine();
82
83                  insertGame(name,size,genre);
84                  break;
85
86              case 2:
87                  showGames();
88                  break;
89
90              case 3:
91                  System.out.println("Enter game name to delete: ");
92                  deleteGame(reader.readLine());
93                  break;
94
95              case 4:
96                  reader.close();
97                  System.exit(0);
98                  break;
99
100             default:
101                 System.out.println("Wrong input , please try again!");
102             }
103
```

MongoCollection.java

```
                    insertGame(name,size,genre);
                    break;

                case 2:
                    showGames();
                    break;

                case 3:
                    System.out.println("Enter game name to delete: ");
                    deleteGame(reader.readLine());
                    break;

                case 4:
                    reader.close();
                    System.exit(0);
                    break;

                default:
                    System.out.println("Wrong input , please try again!");
                }

                System.out.println();
            }
        }
            catch(Exception e) {
                e.printStackTrace();
            }
        }
    }


```

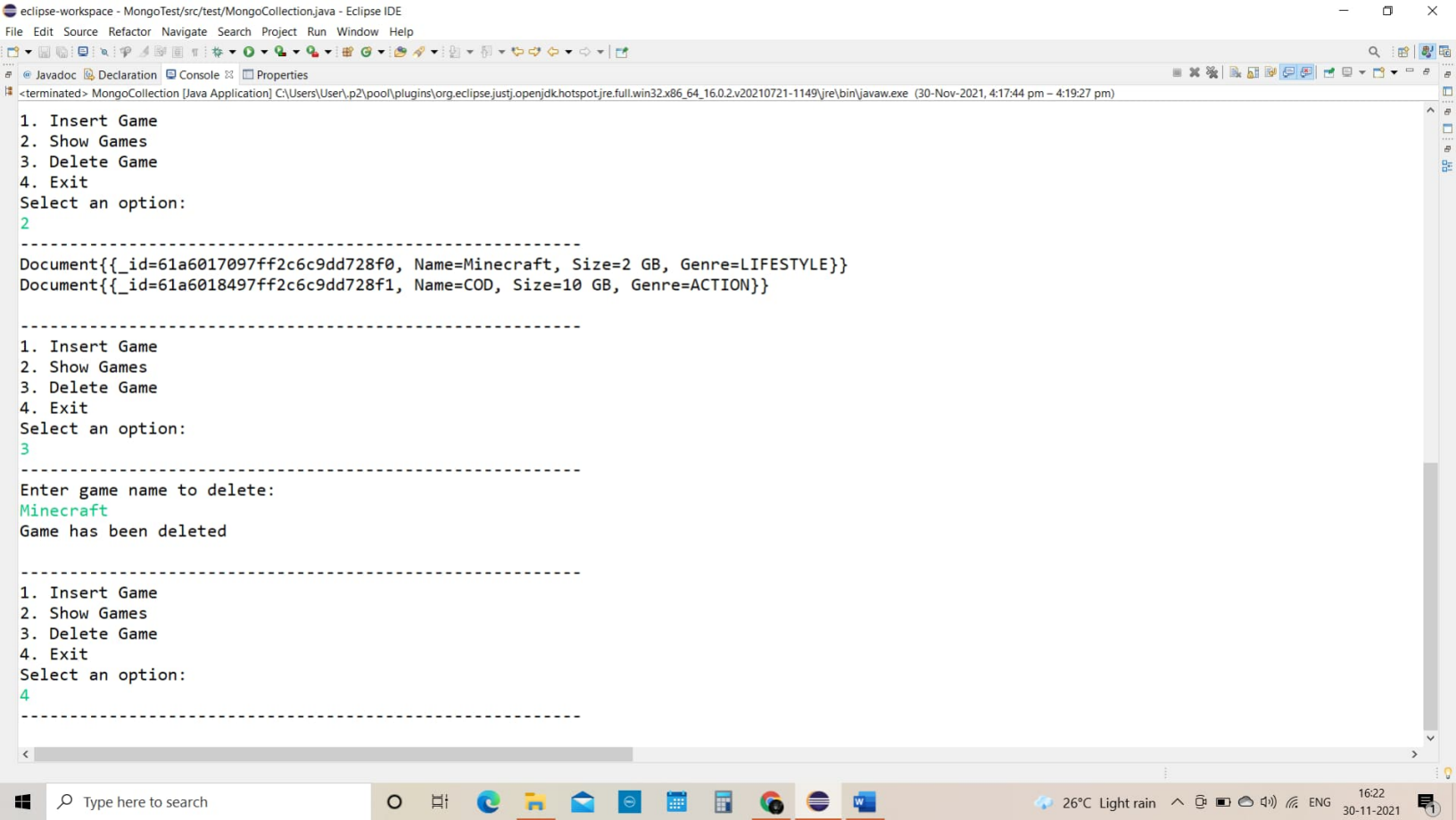

eclipse-workspace - MongoTest/src/test/MongoCollection.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Javadoc Declaration Console Properties
<terminated> MongoCollection [Java Application] C:\Users\User\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jre\bin\javaw.exe (30-Nov-2021, 4:17:44 pm – 4:19:27 pm)
Nov 30, 2021 4:17:44 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[127.0.0.1:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', max
1. Insert Game
2. Show Games
3. Delete Game
4. Exit
Select an option:
Nov 30, 2021 4:17:44 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:26}] to 127.0.0.1:27017
Nov 30, 2021 4:17:44 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=127.0.0.1:27017, type=STANDALONE, state=CONNE
1
Enter game name:
Minecraft
Enter game size:
2
Enter game genre:
Lifestyle
Nov 30, 2021 4:18:16 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:27}] to 127.0.0.1:27017
Inserted Successfully
1. Insert Game
2. Show Games
3. Delete Game
4. Exit
Select an option:
1
Type here to search
26°C Light rain
ENG
16:22
30-11-2021


eclipse-workspace - MongoTest/src/test/MongoCollection.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Javadoc Declaration Console Properties
<terminated> MongoCollection [Java Application] C:\Users\User\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jre\bin\javaw.exe (30-Nov-2021, 4:17:44 pm – 4:19:27 pm)
INFO: Opened connection [connectionId{localValue:2, serverValue:27}] to 127.0.0.1:27017
Inserted Successfully
---
1. Insert Game
2. Show Games
3. Delete Game
4. Exit
Select an option:
1

Enter game name:
COD
Enter game size:
10
Enter game genre:
Action
Inserted Successfully
---
1. Insert Game
2. Show Games
3. Delete Game
4. Exit
Select an option:
2

Document{{_id=61a6017097ff2c6c9dd728f0, Name=Minecraft, Size=2 GB, Genre=LIFESTYLE}}
Document{{_id=61a6018497ff2c6c9dd728f1, Name=COD, Size=10 GB, Genre=ACTION}}

Type here to search
26°C Light rain
ENG
16:22
30-11-2021


eclipse-workspace - MongoTest/src/test/MongoCollection.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Javadoc Declaration Console Properties
<terminated> MongoCollection [Java Application] C:\Users\User\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jre\bin\javaw.exe (30-Nov-2021, 4:17:44 pm – 4:19:27 pm)
1. Insert Game
2. Show Games
3. Delete Game
4. Exit
Select an option:
2
Document{{_id=61a6017097ff2c6c9dd728f0, Name=Minecraft, Size=2 GB, Genre=LIFESTYLE}}
Document{{_id=61a6018497ff2c6c9dd728f1, Name=COD, Size=10 GB, Genre=ACTION}}
1. Insert Game
2. Show Games
3. Delete Game
4. Exit
Select an option:
3
Enter game name to delete:
Minecraft
Game has been deleted
1. Insert Game
2. Show Games
3. Delete Game
4. Exit
Select an option:
4
Type here to search
26°C Light rain
ENG
16:22
30-11-2021