

Relational Databases with MySQL Week 10 Assignment

Points possible: 70

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

Instructions: In Eclipse, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document to the repository. Additionally, push an .sql file with all your queries and your Java project code to the same repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Coding Steps:

In this week's coding activity, you will create a menu driven application backed by a MySQL database.

To start, choose one item that you like. It could be vehicles, sports, foods, etc....

Create a new Java project in Eclipse.

Create a SQL script in the project to create a database with one table. The table should be the item you picked.

Write a Java menu driven application that allows you to perform all four CRUD operations on your table.

Tips:

The application does not need to be as complex as the example in the video curriculum.

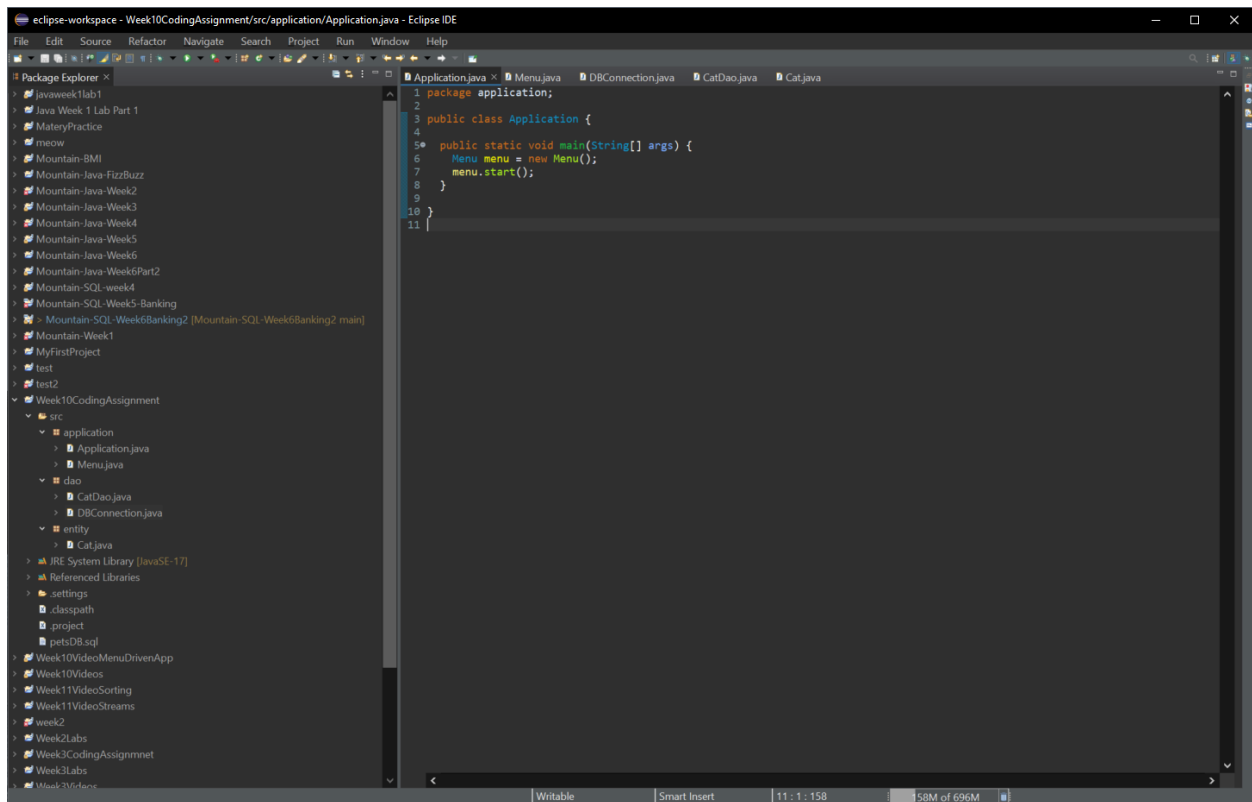
You need an option for each of the CRUD operations (Create, Read, Update, and Delete).

Remember that `PreparedStatement.executeQuery()` is only for Reading data and `.executeUpdate()` is used for Creating, Updating, and Deleting data.

Remember that both parameters on `PreparedStatement`s and the `ResultSet` columns are based on indexes that start with 1, not 0.

Screenshots of Code:

Application:

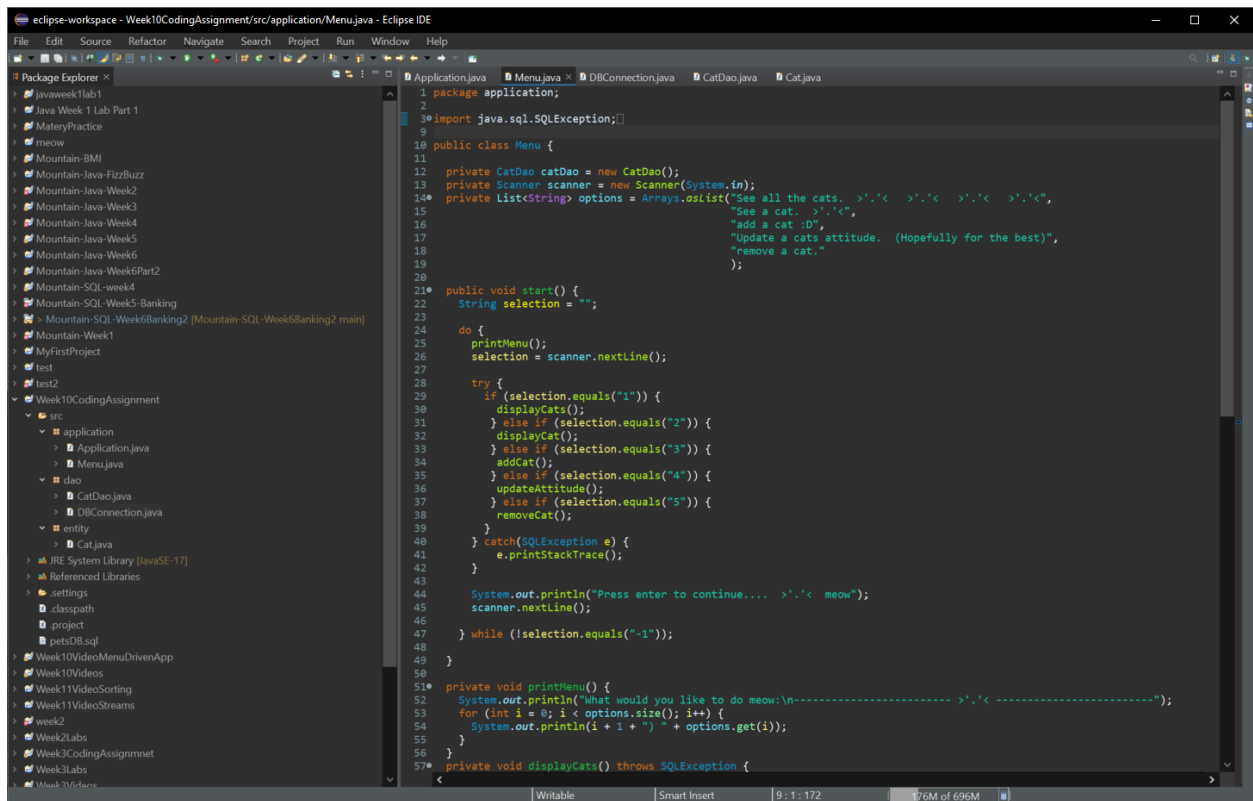


The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays a project structure with a package named 'application' containing 'Application.java'. The main editor window shows the code for 'Application.java'.

```
1 package application;
2
3 public class Application {
4
5     public static void main(String[] args) {
6         Menu menu = new Menu();
7         menu.start();
8     }
9 }
10
11 }
```

The status bar at the bottom indicates the file is 'Writable', 'Smart Insert' is active, the cursor is at line 11, column 158, and the file size is 58M of 696M.

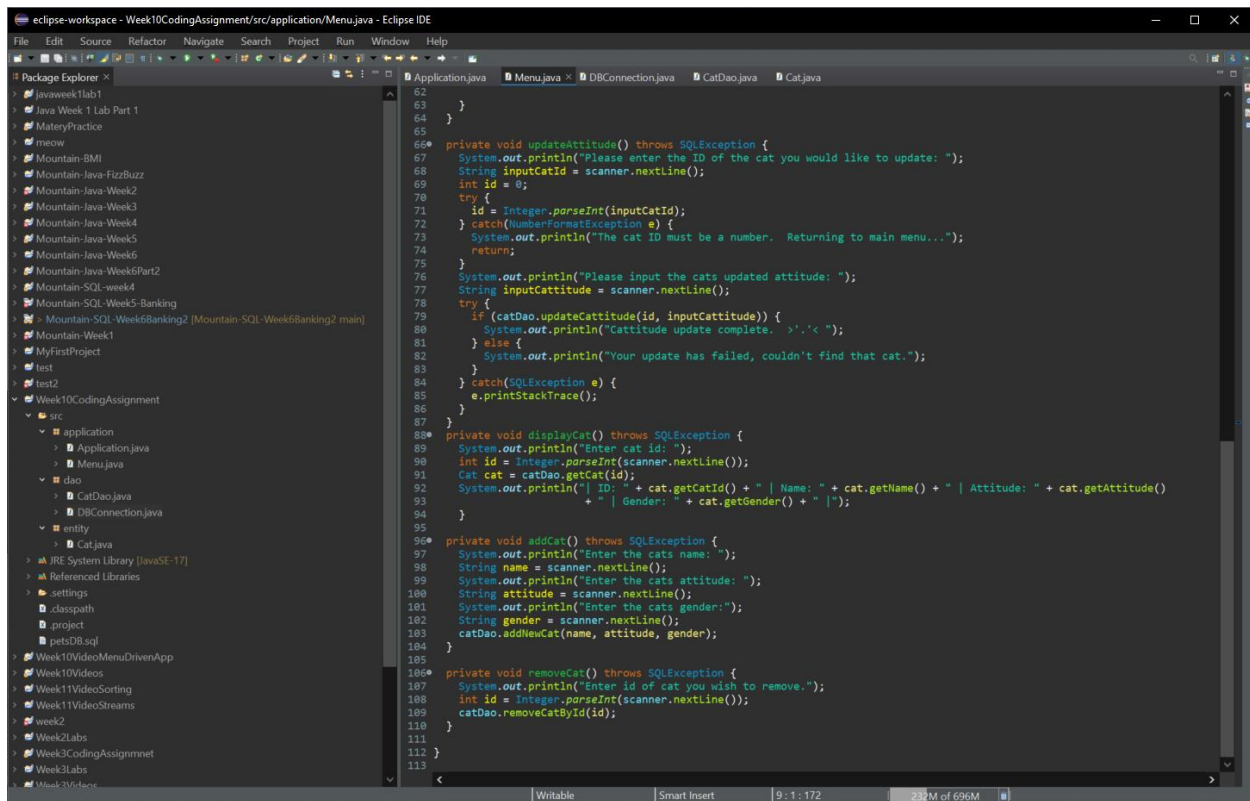
Menu 1/2:



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays a project named 'Week10CodingAssignment' with a package structure including 'application', 'dao', and 'entity'. The main editor window shows the 'Menu.java' file. The code defines a 'Menu' class with a 'start()' method that handles user input for displaying, adding, updating, or removing cats. It also includes a 'printMenu()' method to display the available options.

```
1 package application;
2
3 import java.sql.SQLException;
4
5
6
7
8
9
10 public class Menu {
11
12     private CatDao catDao = new CatDao();
13     private Scanner scanner = new Scanner(System.in);
14     private List<String> options = Arrays.asList("See all the cats. >'.< >'.< >'.< >'.<",
15         "See a cat. >'.<",
16         "add a cat :D",
17         "Update a cats attitude. (Hopefully for the best)",
18         "remove a cat."
19     );
20
21     public void start() {
22         String selection = "";
23
24         do {
25             printMenu();
26             selection = scanner.nextLine();
27
28             try {
29                 if (selection.equals("1")) {
30                     displayCats();
31                 } else if (selection.equals("2")) {
32                     displayCat();
33                 } else if (selection.equals("3")) {
34                     addCat();
35                 } else if (selection.equals("4")) {
36                     updateAttitude();
37                 } else if (selection.equals("5")) {
38                     removeCat();
39                 }
40             } catch (SQLException e) {
41                 e.printStackTrace();
42             }
43
44             System.out.println("Press enter to continue.... >'.< meow");
45             scanner.nextLine();
46
47             while (!selection.equals("-1"));
48         }
49     }
50
51     private void printMenu() {
52         System.out.println("What would you like to do meow:\n----- >'.< -----");
53         for (int i = 0; i < options.size(); i++) {
54             System.out.println(i + 1 + " " + options.get(i));
55         }
56     }
57     private void displayCats() throws SQLException {
```

Menu 2/2:

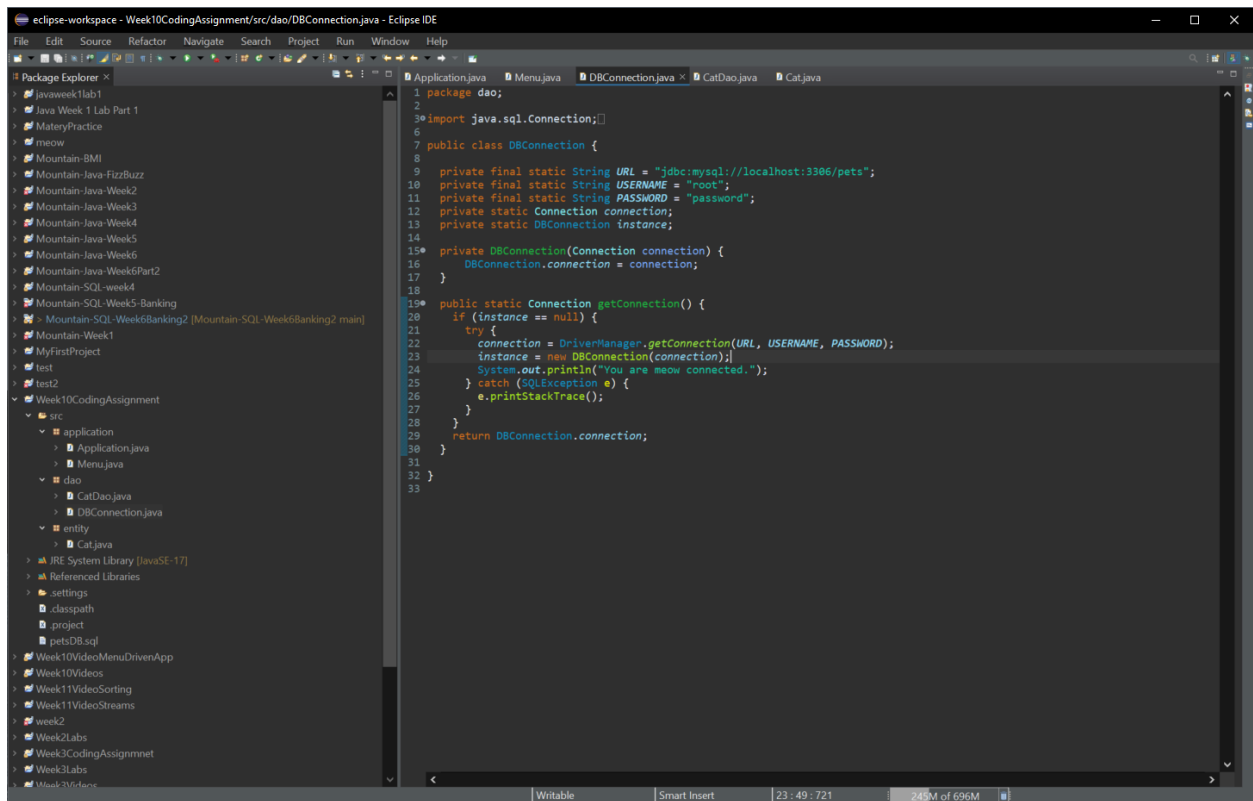


The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays the project structure for 'Week10CodingAssignment', including packages like 'application', 'dao', and 'entity'. The main editor window shows the 'Menu.java' file with the following code:

```
62 }
63
64 }
65
66 private void updateAttitude() throws SQLException {
67     System.out.println("Please enter the ID of the cat you would like to update: ");
68     String inputCatId = scanner.nextLine();
69     int id = 0;
70     try {
71         id = Integer.parseInt(inputCatId);
72     } catch (NumberFormatException e) {
73         System.out.println("The cat ID must be a number. Returning to main menu...");
74         return;
75     }
76     System.out.println("Please input the cats updated attitude: ");
77     String inputCatAttitude = scanner.nextLine();
78     try {
79         if (catDao.updateCatAttitude(id, inputCatAttitude)) {
80             System.out.println("Cat attitude update complete. >.'< ");
81         } else {
82             System.out.println("Your update has failed, couldn't find that cat.");
83         }
84     } catch (SQLException e) {
85         e.printStackTrace();
86     }
87 }
88 private void displayCat() throws SQLException {
89     System.out.println("Enter cat id: ");
90     int id = Integer.parseInt(scanner.nextLine());
91     Cat cat = catDao.getCat(id);
92     System.out.println("ID: " + cat.getCatId() + " | Name: " + cat.getName() + " | Attitude: " + cat.getAttitude()
93         + " | Gender: " + cat.getGender() + " |");
94 }
95
96 private void addCat() throws SQLException {
97     System.out.println("Enter the cats name: ");
98     String name = scanner.nextLine();
99     System.out.println("Enter the cats attitude: ");
100    String attitude = scanner.nextLine();
101    System.out.println("Enter the cats gender:");
102    String gender = scanner.nextLine();
103    catDao.addNewCat(name, attitude, gender);
104 }
105
106 private void removeCat() throws SQLException {
107     System.out.println("Enter id of cat you wish to remove.");
108     int id = Integer.parseInt(scanner.nextLine());
109     catDao.removeCatById(id);
110 }
111
112 }
113 }
```

The status bar at the bottom indicates the file is 'Writable', 'Smart Insert' is active, and the cursor is at line 9, column 172. The memory usage is shown as 22M of 696M.

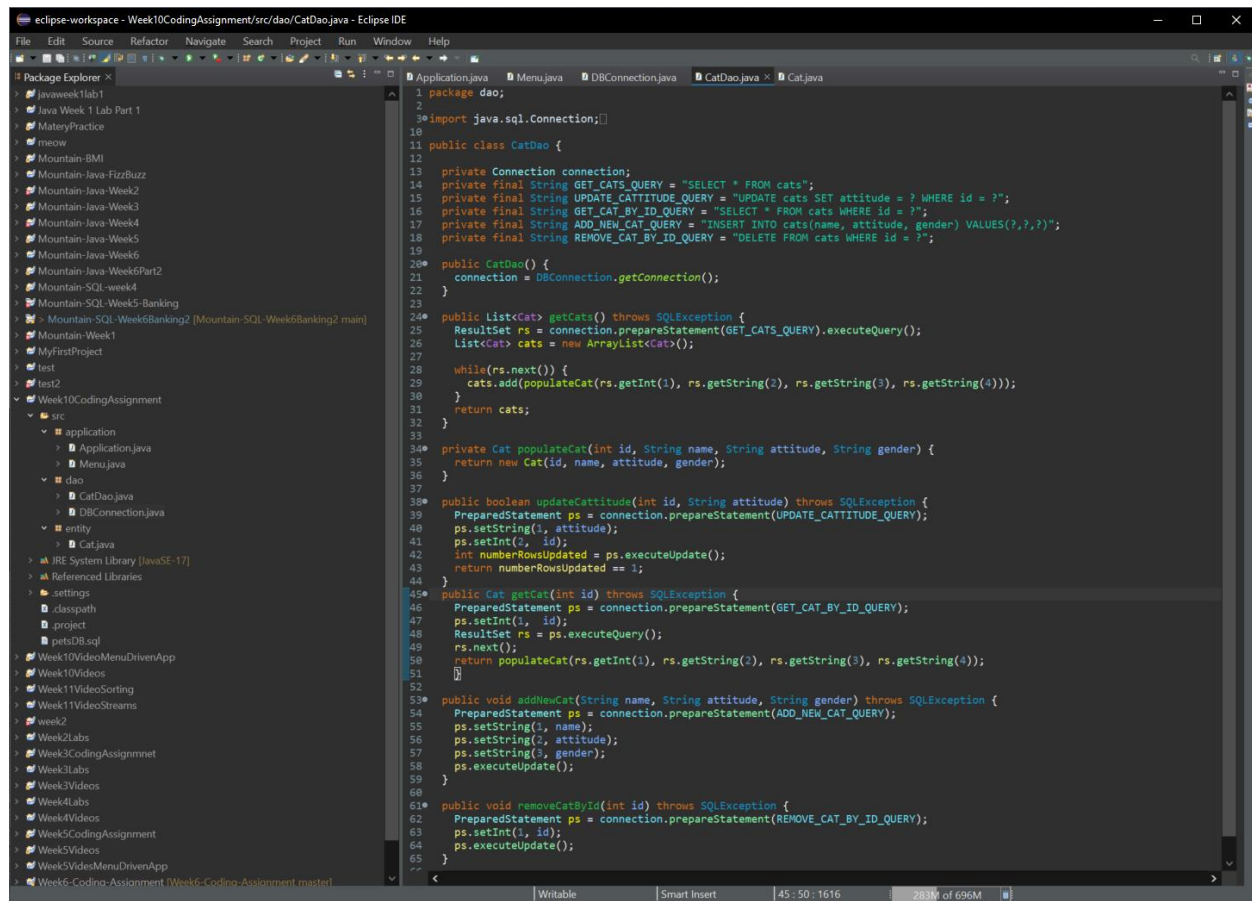
DBConnection:



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays a project structure with a 'dao' package containing 'DBConnection.java'. The main editor window shows the code for 'DBConnection.java'.

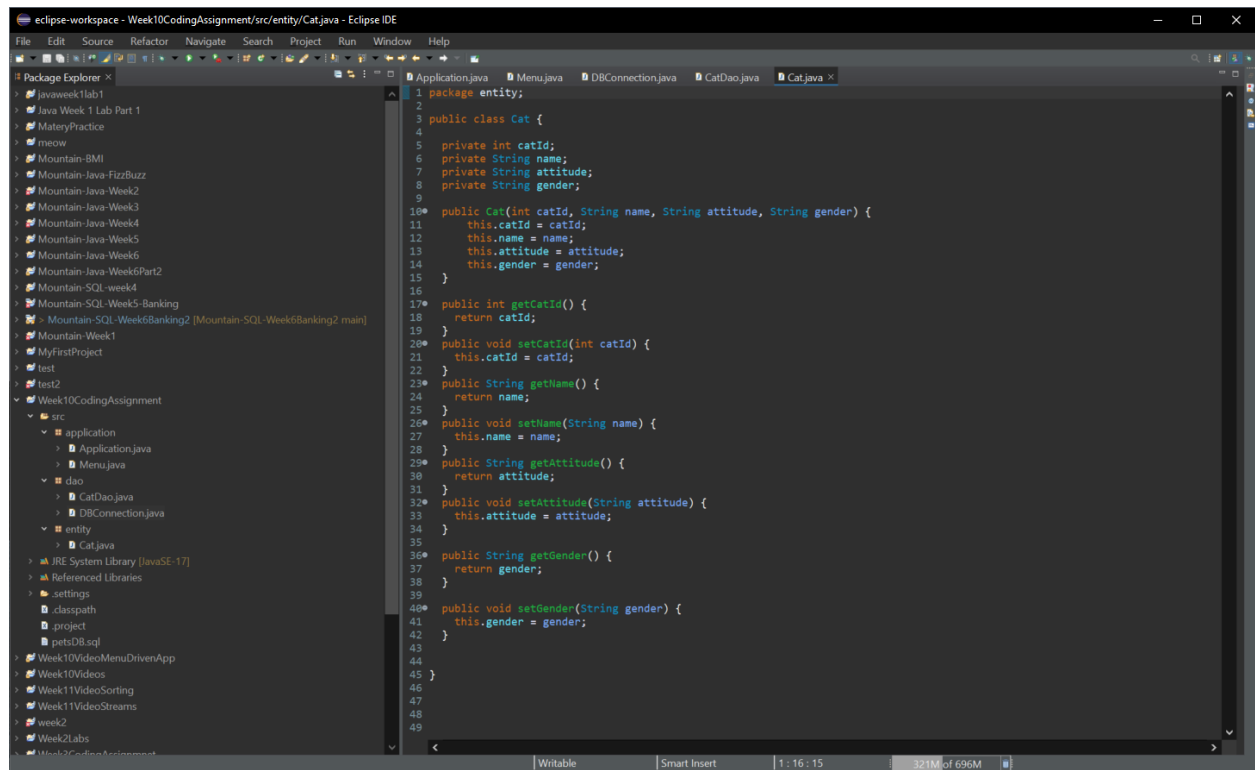
```
1 package dao;
2
3 import java.sql.Connection;
4
5
6
7 public class DBConnection {
8
9     private final static String URL = "jdbc:mysql://localhost:3306/pets";
10    private final static String USERNAME = "root";
11    private final static String PASSWORD = "password";
12    private static Connection connection;
13    private static DBConnection instance;
14
15    private DBConnection(Connection connection) {
16        DBConnection.connection = connection;
17    }
18
19    public static Connection getConnection() {
20        if (instance == null) {
21            try {
22                connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
23                instance = new DBConnection(connection);
24                System.out.println("You are meow connected.");
25            } catch (SQLException e) {
26                e.printStackTrace();
27            }
28        }
29        return DBConnection.connection;
30    }
31 }
32
33
```

CatDao:



```
1 package dao;
2
3 import java.sql.Connection;
4
5 public class CatDao {
6
7     private Connection connection;
8     private final String GET_CATS_QUERY = "SELECT * FROM cats";
9     private final String UPDATE_CATTITUDE_QUERY = "UPDATE cats SET attitude = ? WHERE id = ?";
10    private final String GET_CAT_BY_ID_QUERY = "SELECT * FROM cats WHERE id = ?";
11    private final String ADD_NEW_CAT_QUERY = "INSERT INTO cats(name, attitude, gender) VALUES(?,?,?)";
12    private final String REMOVE_CAT_BY_ID_QUERY = "DELETE FROM cats WHERE id = ?";
13
14    public CatDao() {
15        connection = DBConnection.getConnection();
16    }
17
18    public List<Cat> getCats() throws SQLException {
19        ResultSet rs = connection.prepareStatement(GET_CATS_QUERY).executeQuery();
20        List<Cat> cats = new ArrayList<Cat>();
21
22        while(rs.next()) {
23            cats.add(populateCat(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getString(4)));
24        }
25        return cats;
26    }
27
28    private Cat populateCat(int id, String name, String attitude, String gender) {
29        return new Cat(id, name, attitude, gender);
30    }
31
32    public boolean updateCattitude(int id, String attitude) throws SQLException {
33        PreparedStatement ps = connection.prepareStatement(UPDATE_CATTITUDE_QUERY);
34        ps.setString(1, attitude);
35        ps.setInt(2, id);
36        int numberOfRowsUpdated = ps.executeUpdate();
37        return numberOfRowsUpdated == 1;
38    }
39
40    public Cat getCat(int id) throws SQLException {
41        PreparedStatement ps = connection.prepareStatement(GET_CAT_BY_ID_QUERY);
42        ps.setInt(1, id);
43        ResultSet rs = ps.executeQuery();
44        rs.next();
45        return populateCat(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getString(4));
46    }
47
48    public void addNewCat(String name, String attitude, String gender) throws SQLException {
49        PreparedStatement ps = connection.prepareStatement(ADD_NEW_CAT_QUERY);
50        ps.setString(1, name);
51        ps.setString(2, attitude);
52        ps.setString(3, gender);
53        ps.executeUpdate();
54    }
55
56    public void removeCatById(int id) throws SQLException {
57        PreparedStatement ps = connection.prepareStatement(REMOVE_CAT_BY_ID_QUERY);
58        ps.setInt(1, id);
59        ps.executeUpdate();
60    }
61
62    ...
63 }
```

Cat:



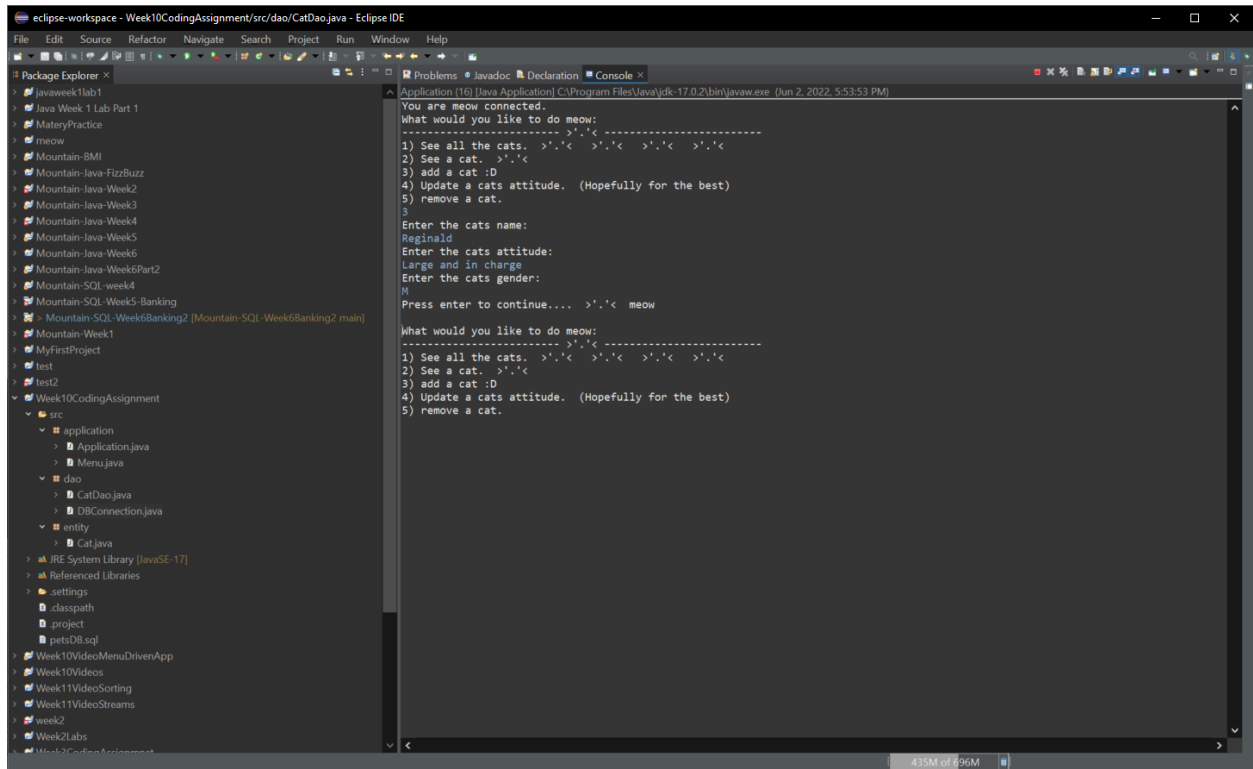
The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays a project structure with a package named 'entity' containing a file 'Cat.java'. The main editor window shows the code for 'Cat.java'.

```
1 package entity;
2
3 public class Cat {
4
5     private int catId;
6     private String name;
7     private String attitude;
8     private String gender;
9
10    public Cat(int catId, String name, String attitude, String gender) {
11        this.catId = catId;
12        this.name = name;
13        this.attitude = attitude;
14        this.gender = gender;
15    }
16
17    public int getCatId() {
18        return catId;
19    }
20    public void setCatId(int catId) {
21        this.catId = catId;
22    }
23    public String getName() {
24        return name;
25    }
26    public void setName(String name) {
27        this.name = name;
28    }
29    public String getAttitude() {
30        return attitude;
31    }
32    public void setAttitude(String attitude) {
33        this.attitude = attitude;
34    }
35
36    public String getGender() {
37        return gender;
38    }
39
40    public void setGender(String gender) {
41        this.gender = gender;
42    }
43
44 }
45
46
47
48
49
```

The status bar at the bottom indicates the file is 'Writable', 'Smart Insert' is active, the cursor is at '1 : 16 : 15', and the memory usage is '321M of 696M'.

Screenshots of Running Application:

Create:



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays a project structure with a package named 'dao' containing 'CatDao.java'. The main editor area shows the 'Console' output, which displays the application's execution. The application is titled 'Application (16) [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\java.exe (Jun 2, 2022, 5:53:53 PM)'. The console output shows the application is connected to a database and displays a menu of options for interacting with cats. The user has entered 'Reginaid' for the cat's name, 'Large and in charge' for the cat's attitude, and 'M' for the cat's gender. The application then prompts the user to press enter to continue, which they do, resulting in the output 'meow'.

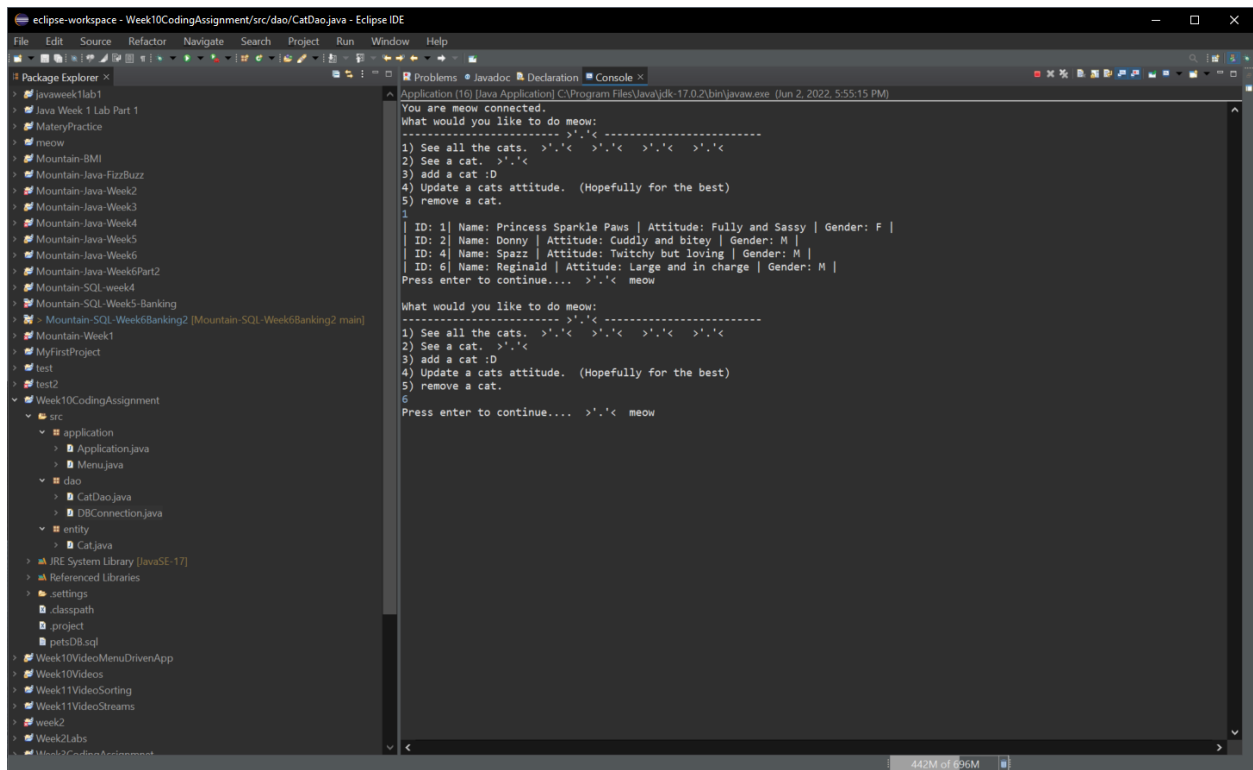
```
eclipse-workspace - Week10CodingAssignment/src/dao/CatDao.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer x
  javawork1lab1
  Java Week 1 Lab Part 1
  MateryPractice
  meow
  Mountain-BMI
  Mountain-Java-FizzBuzz
  Mountain-Java-Week2
  Mountain-Java-Week3
  Mountain-Java-Week4
  Mountain-Java-Week5
  Mountain-Java-Week6
  Mountain-Java-Week6Part2
  Mountain-SQL-Week4
  Mountain-SQL-Week5-Banking
  Mountain-SQL-Week6Banking2 [Mountain-SQL-Week6Banking2 main]
  Mountain-Week1
  MyFirstProject
  test
  test2
  Week10CodingAssignment
    src
      application
        Application.java
        Menu.java
      dao
        CatDao.java
        DBConnection.java
      entity
        Cat.java
  JRE System Library [JavaSE-17]
  Referenced Libraries
  settings
  classpath
  project
  petsDB.sql
  Week10VideoMenuDrivenApp
  Week10Videos
  Week11VideoSorting
  Week11VideoStreams
  week2
  Week2labs
  Week2CodingAssignment

Problems Javadoc Declaration Console x
Application (16) [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\java.exe (Jun 2, 2022, 5:53:53 PM)
You are meow connected.
What would you like to do meow:
----->'< >'< >'< >'< >'<
1) See all the cats. >'< >'< >'< >'< >'<
2) See a cat. >'<
3) add a cat :D
4) Update a cats attitude. (Hopefully for the best)
5) remove a cat.
3
Enter the cats name:
Reginaid
Enter the cats attitude:
Large and in charge
Enter the cats gender:
M
Press enter to continue.... >'< meow

What would you like to do meow:
----->'< >'< >'< >'< >'<
1) See all the cats. >'< >'< >'< >'< >'<
2) See a cat. >'<
3) add a cat :D
4) Update a cats attitude. (Hopefully for the best)
5) remove a cat.
4
```


Read:



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays a project structure with a package named 'dao' containing 'CatDao.java'. The Console window on the right shows the output of a Java application. The application is titled 'Application (16) [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (Jun 2, 2022, 5:55:15 PM)'. The console output indicates that the application is connected to a database and displays a list of cats. The cats are listed as follows:

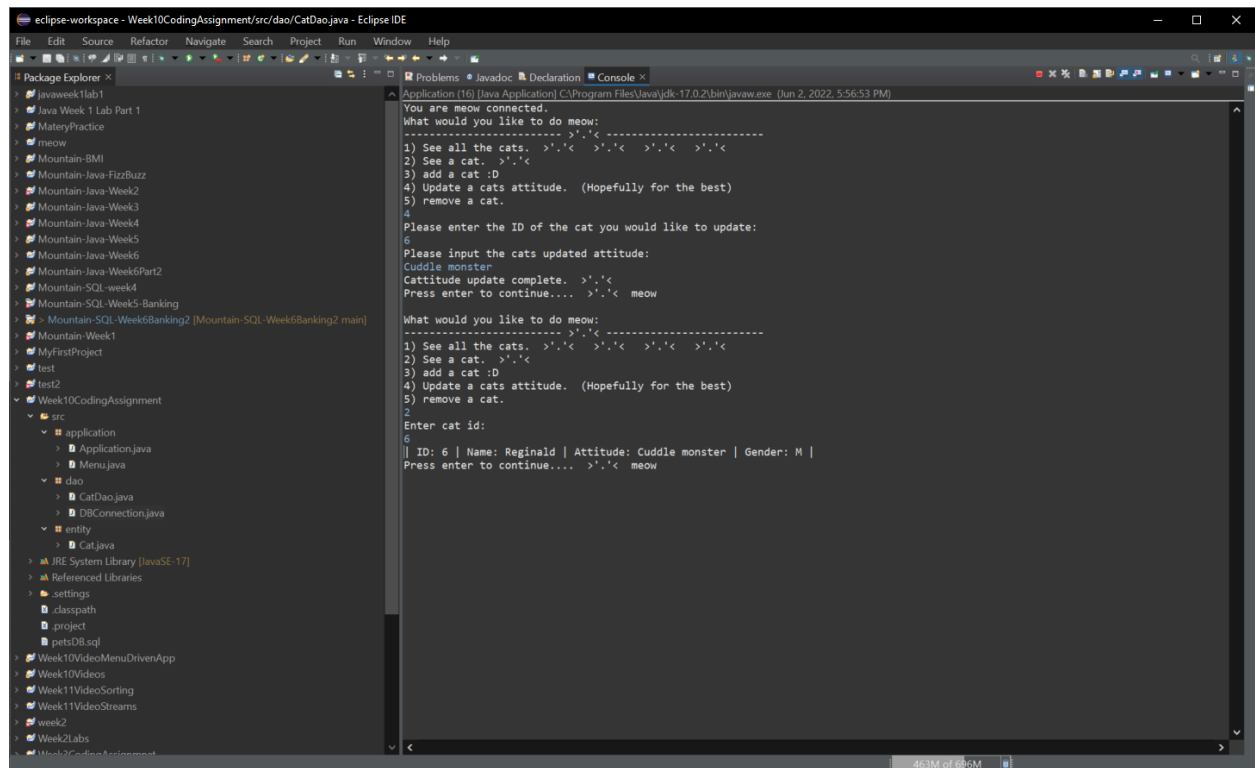
ID	Name	Attitude	Gender
1	Princess Sparkle Paws	Fully and Sassy	F
2	Donny	Cuddly and bitey	M
4	Spazz	Twitchy but loving	M
6	Reginald	Large and in charge	M

The application also displays a menu with the following options:

- 1) See all the cats.
- 2) See a cat.
- 3) add a cat :D
- 4) Update a cats attitude. (Hopefully for the best)
- 5) remove a cat.

The application prompts the user to 'Press enter to continue...' and the user has entered 'meow'.

Update:

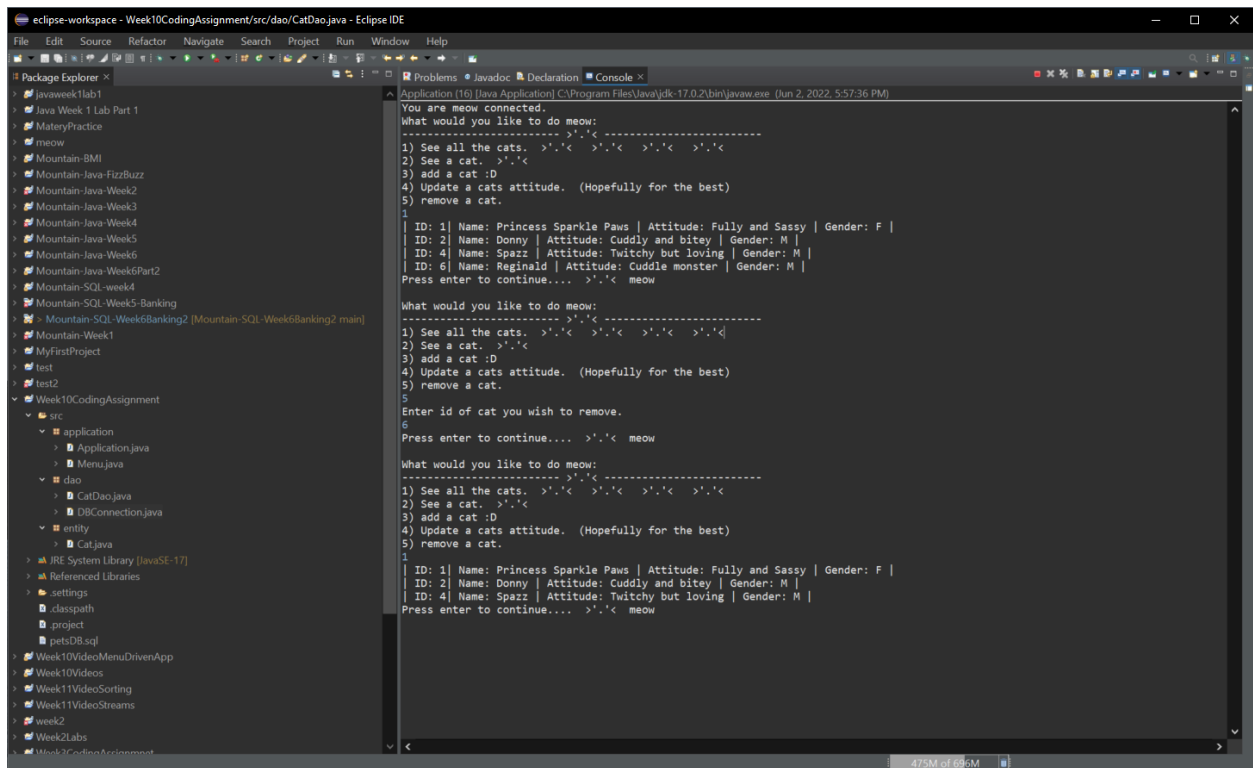


```
eclipse-workspace - Week10CodingAssignment/src/dao/CatDao.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer x
  javaWeek1Lab1
  Java Week 1 Lab Part 1
  MateryPractice
  meow
  Mountain-BMI
  Mountain-Java-FizzBuzz
  Mountain-Java-Week2
  Mountain-Java-Week3
  Mountain-Java-Week4
  Mountain-Java-Week5
  Mountain-Java-Week6
  Mountain-Java-Week6Part2
  Mountain-SQL-week4
  Mountain-SQL-Week5-Banking
  Mountain-SQL-Week6Banking2 [Mountain-SQL-Week6Banking2.main]
  Mountain-Week1
  MyFirstProject
  test
  test2
  Week10CodingAssignment
    src
      application
        Application.java
        Menu.java
      dao
        CatDao.java
        DBConnection.java
      entity
        Cat.java
  JRE System Library [JavaSE-17]
  Referenced Libraries
  settings
    classpath
    project
    petsDB.sql
  Week10VideoMenuDrivenApp
  Week10Videos
  Week11VideoSorting
  Week11VideoStreams
  week2
  Week21labs
  Week2CodingAssignment

Problems Javadoc Declaration Console x
Application (16) [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (Jun 2, 2022, 5:56:53 PM)
You are meow connected.
What would you like to do meow:
----- >','< -----
1) See all the cats. >','< >','< >','< >','<
2) See a cat. >','<
3) add a cat :D
4) Update a cats attitude. (Hopefully for the best)
5) remove a cat.
6
Please enter the ID of the cat you would like to update:
6
Please input the cats updated attitude:
Cuddle monster
Attitude update complete. >','<
Press enter to continue.... >','< meow

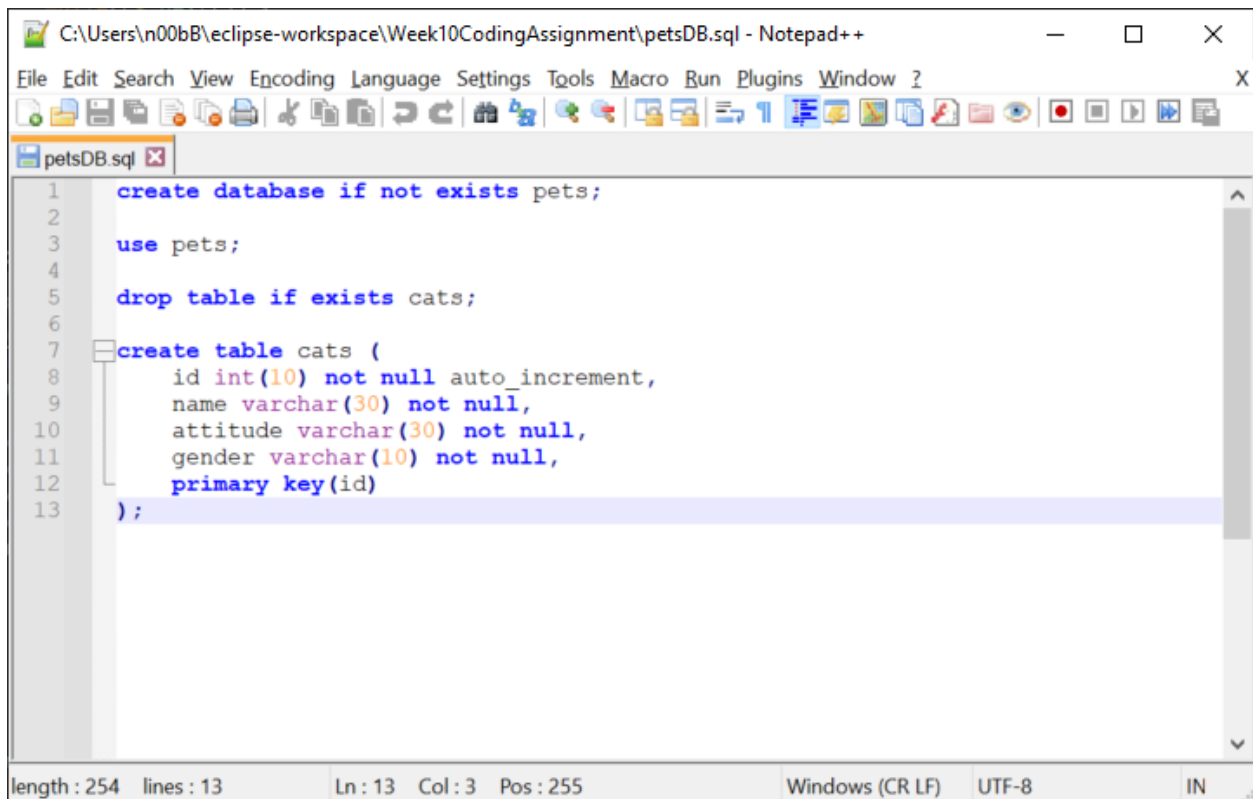
What would you like to do meow:
----- >','< -----
1) See all the cats. >','< >','< >','< >','<
2) See a cat. >','<
3) add a cat :D
4) Update a cats attitude. (Hopefully for the best)
5) remove a cat.
6
Enter cat id:
6
|| ID: 6 | Name: Reginald | Attitude: Cuddle monster | Gender: M |
Press enter to continue.... >','< meow
```

Delete:



```
eclipse-workspace - Week10CodingAssignment/src/dao/CatDao.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
javaWeek1Lab1
Java Week 1 Lab Part 1
MateryPractice
meow
Mountain-BMI
Mountain-Java-FizzBuzz
Mountain-Java-Week2
Mountain-Java-Week3
Mountain-Java-Week4
Mountain-Java-Week5
Mountain-Java-Week6
Mountain-Java-Week6Part2
Mountain-SQL-week4
Mountain-SQL-Week5-Banking
Mountain-SQL-Week6Banking2 [Mountain-SQL-Week6Banking2 main]
Mountain-Week1
MyFirstProject
test
test2
Week10CodingAssignment
src
application
Application.java
Menu.java
dao
CatDao.java
DBConnection.java
entity
Cat.java
JRE System Library [javaSE-17]
Referenced Libraries
settings
classpath
project
petsDB.sql
Week10VideoMenuDrivenApp
Week10Videos
Week11VideoSorting
Week11VideoStreams
week2
Week2Labs
Week3CodingAssignment
Problems Javadoc Declaration Console
Application [16] [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (Jun 2, 2022, 5:57:36 PM)
You are meow connected.
What would you like to do meow:
----->'.'<
1) See all the cats. >'.'< >'.'< >'.'< >'.'<
2) See a cat. >'.'<
3) add a cat :D
4) Update a cats attitude. (Hopefully for the best)
5) remove a cat.
1
ID: 1 Name: Princess Sparkle Paws | Attitude: Fully and Sassy | Gender: F |
ID: 2 Name: Donny | Attitude: Cuddly and bitey | Gender: M |
ID: 4 Name: Spazz | Attitude: Twitchy but loving | Gender: M |
ID: 6 Name: Reginald | Attitude: Cuddle monster | Gender: M |
Press enter to continue.... >'.'< meow
What would you like to do meow:
----->'.'<
1) See all the cats. >'.'< >'.'< >'.'< >'.'<
2) See a cat. >'.'<
3) add a cat :D
4) Update a cats attitude. (Hopefully for the best)
5) remove a cat.
5
Enter id of cat you wish to remove.
6
Press enter to continue.... >'.'< meow
What would you like to do meow:
----->'.'<
1) See all the cats. >'.'< >'.'< >'.'< >'.'<
2) See a cat. >'.'<
3) add a cat :D
4) Update a cats attitude. (Hopefully for the best)
5) remove a cat.
5
ID: 1 Name: Princess Sparkle Paws | Attitude: Fully and Sassy | Gender: F |
ID: 2 Name: Donny | Attitude: Cuddly and bitey | Gender: M |
ID: 4 Name: Spazz | Attitude: Twitchy but loving | Gender: M |
Press enter to continue.... >'.'< meow
```

petsDB.sql:



```
C:\Users\n00bB\eclipse-workspace\Week10CodingAssignment\petsDB.sql - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
petsDB.sql
1 create database if not exists pets;
2
3 use pets;
4
5 drop table if exists cats;
6
7 create table cats (
8     id int(10) not null auto_increment,
9     name varchar(30) not null,
10    attitude varchar(30) not null,
11    gender varchar(10) not null,
12    primary key(id)
13 );
length: 254 lines: 13 Ln: 13 Col: 3 Pos: 255 Windows (CR LF) UTF-8 IN
```

URL to GitHub Repository:

<https://github.com/LoganMunsterman/Week10CodingAssignment>