Relational Databases with MySQL Week 4 Coding 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: Complete the coding steps. Take screenshots of the steps 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 the Java project 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:

1. Create a class of whatever type you want (Animal, Person, Camera, Cheese, etc.).
    a. Do not implement the Comparable interface.
    b. Add a name instance variable so that you can tell the objects apart.
    c. Add getters, setters and/or a constructor as appropriate.
    d. Add a toString method that returns the name and object type (like "Pentax Camera").
    e. Create a static method named `compare` in the class that returns an int and takes two of the objects as parameters. Return -1 if parameter 1 is "less than" parameter 2. Return 1 if parameter 1 is "greater than" parameter 2. Return 0 if the two parameters are "equal".

f. Create a static list of these objects, adding at least 4 objects to the list.

g. In another class, write a method to sort the objects using a Lambda expression using the compare method you created earlier.********

h. Write a method to sort the objects using a Method Reference to the compare method you created earlier.

i. Create a main method to call the sort methods.

j. Print the list after sorting (System.out.println).

2. Create a new class with a main method. Using the list of objects you created in the prior step.

    a. Create a Stream from the list of objects.

    b. Turn the Stream of object to a Stream of String (use the map method for this).

    c. Sort the Stream in the natural order. (Note: The String class implements the Comparable interface, so you won't have to supply a Comparator to do the sorting.)

    d. Collect the Stream and return a comma-separated list of names as a single String. Hint: use Collectors.joining(", ") for this.

    e. Print the resulting String.

3. Create a new class with a main method. Create a method (method a) that accepts an Optional of some type of object (Animal, Person, Camera, etc.).

    a. The method should return the object unwrapped from the Optional if the object is present. For example, if you have an object of type Cheese, your method signature should look something like this:

    ```
    public Cheese cheesyMethod(Optional<Cheese> optionalCheese) {...}
    ```

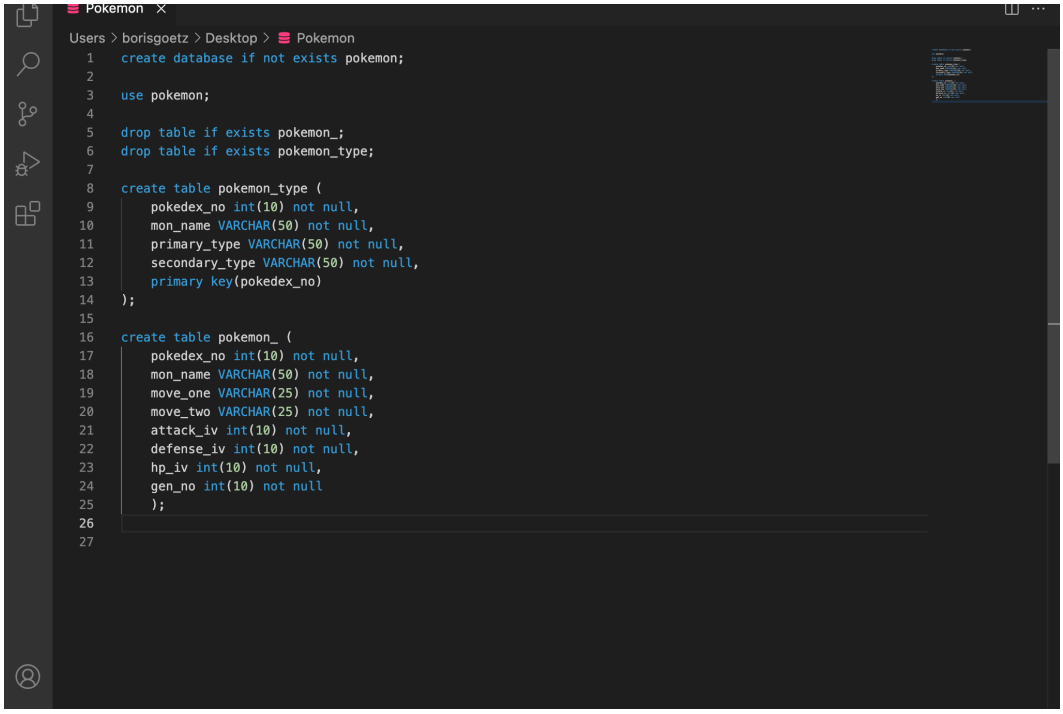
    b. The method should throw a NoSuchElementException with a custom message if the object is not present.

    c. Create another method (method b) that calls method a with an object wrapped by an Optional. Show that the object is returned unwrapped from the Optional (i.e., print the object).

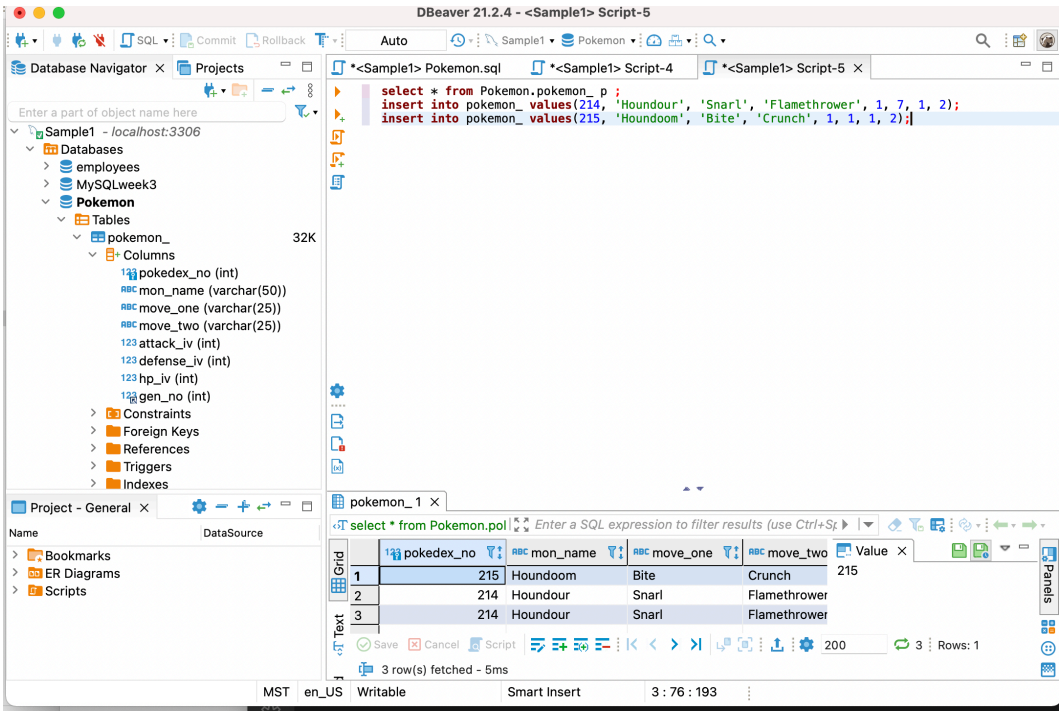
    d. Method b should also call method a with an empty Optional. Show that a NoSuchElementException is thrown by method a by printing the exception message. Hint: catch the `NoSuchElementException` as parameter named "e" and do `System.out.println(e.getMessage())`.

e. Note: your method should handle the Optional as shown in the video on Optionals using the orElseThrow method. For the missing object, you must use a Lambda expression in orElseThrow to return a NoSuchElementException with a custom message.

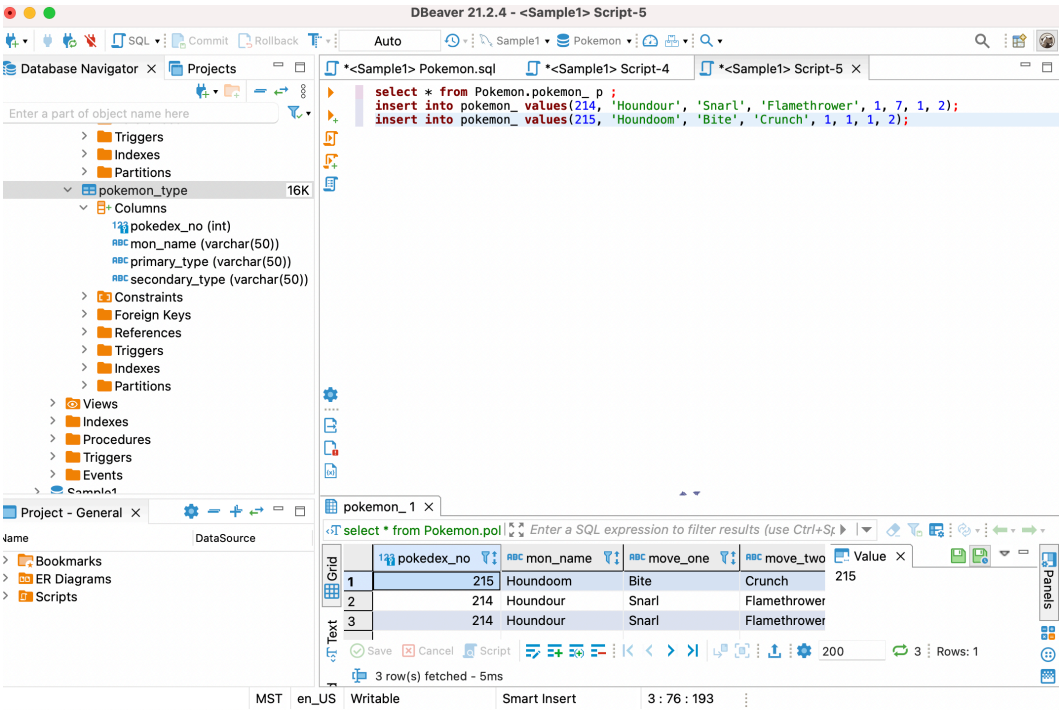
Screenshots:



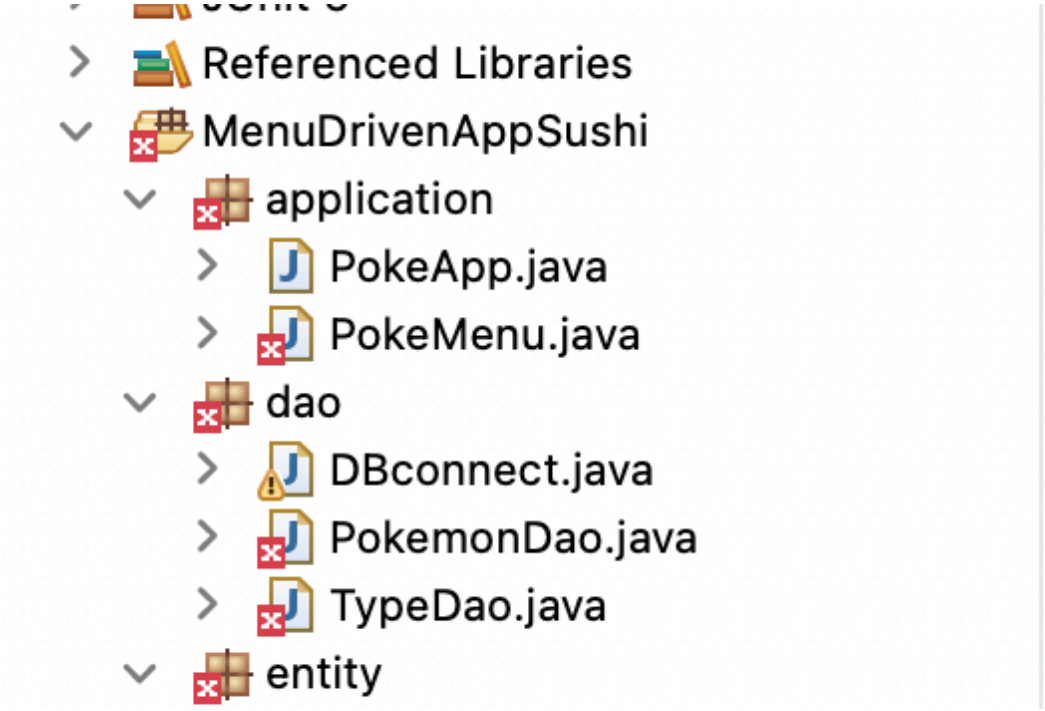
Caption



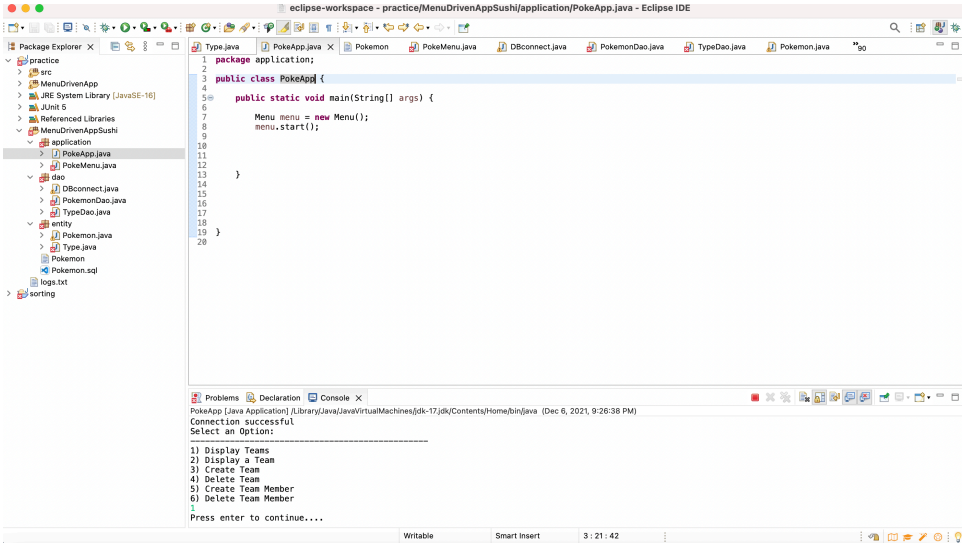
Caption


DBeaver 21.2.4 - <Sample1> Script-5
Database Navigator
Projects
*<Sample1> Pokemon.sql
*<Sample1> Script-4
*<Sample1> Script-5
select * from Pokemon.pokemon_ p ;
insert into pokemon_ values(214, 'Houndour', 'Snarl', 'Flamethrower', 1, 7, 1, 2);
insert into pokemon_ values(215, 'Houndoom', 'Bite', 'Crunch', 1, 1, 1, 2);
Triggers
Indexes
Partitions
pokemon_type
16K
Columns
pokedex_no (int)
mon_name (varchar(50))
primary_type (varchar(50))
secondary_type (varchar(50))
Constraints
Foreign Keys
References
Triggers
Indexes
Partitions
Views
Indexes
Procedures
Triggers
Events
Project - General
Bookmarks
ER Diagrams
Scripts
pokemon_ 1
select * from Pokemon.pol
Enter a SQL expression to filter results (use Ctrl+Sp
pokedex_no
mon_name
move_one
move_two
1 215 Houndoom Bite Crunch
2 214 Houndour Snarl Flamethrower
3 214 Houndour Snarl Flamethrower
Value
215
Grid
Text
Panels
Save
Cancel
Script
200
Rows: 1
3 row(s) fetched - 5ms
MST
en_US
Writable
Smart Insert
3 : 76 : 193

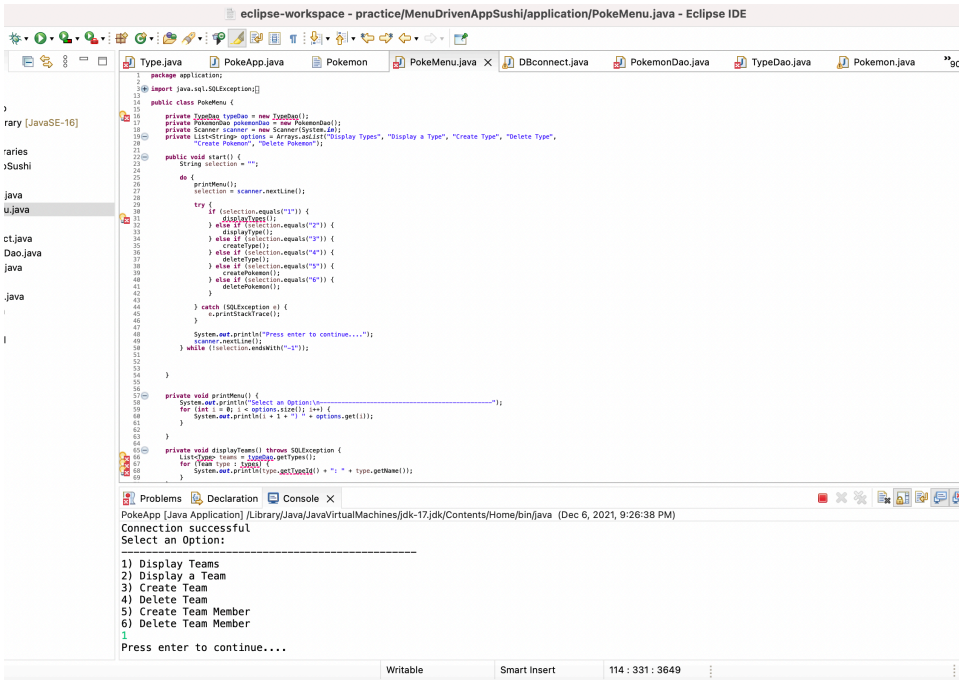
Caption



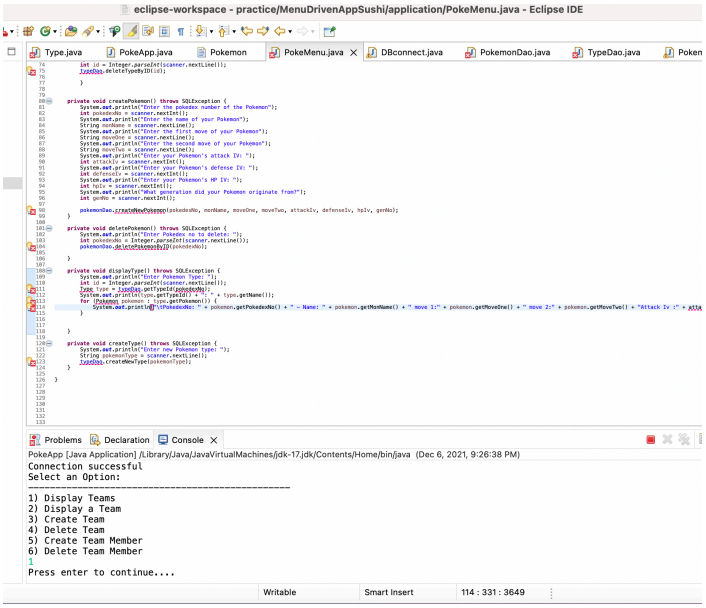
Caption



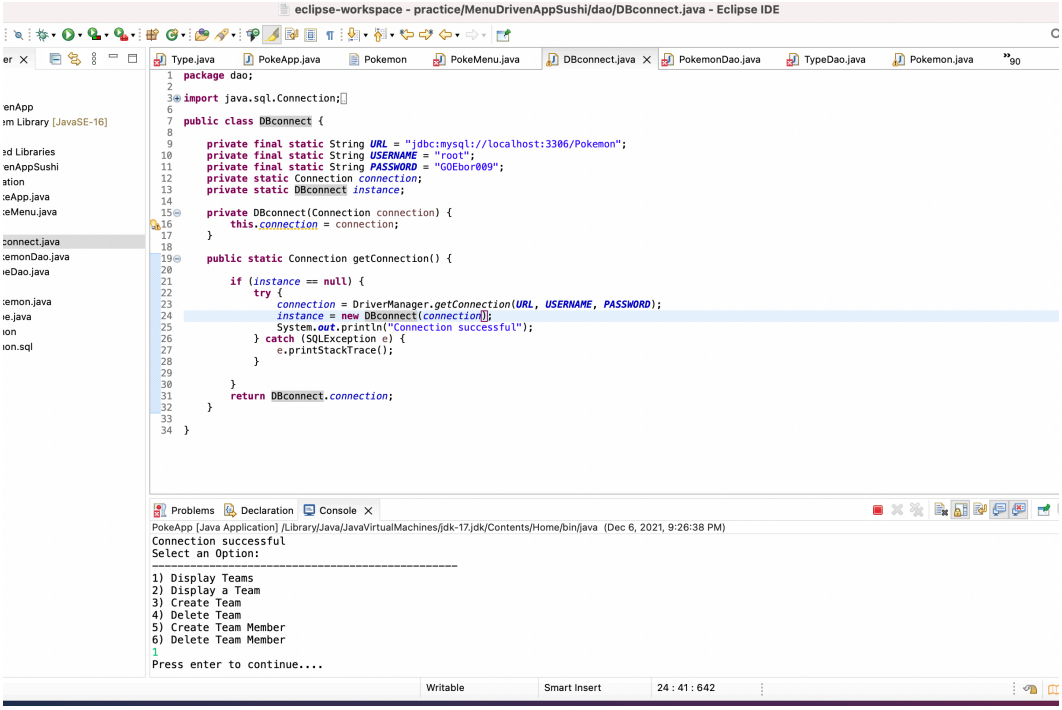
Caption


eclipse-workspace - practice/MenuDrivenAppSushi/application/PokeMenu.java - Eclipse IDE
Type.java
PokeApp.java
Pokemon
PokeMenu.java
DBconnect.java
PokemonDao.java
TypeDao.java
Pokemon.java
Problems
Declaration
Console
PokeApp [Java Application] /Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home/bin/java (Dec 6, 2021, 9:26:38 PM)
Connection successful
Select an Option:
1) Display Teams
2) Display a Team
3) Create Team
4) Delete Team
5) Create Team Member
6) Delete Team Member
1
Press enter to continue....
Writable
Smart Insert
114 : 331 : 3649

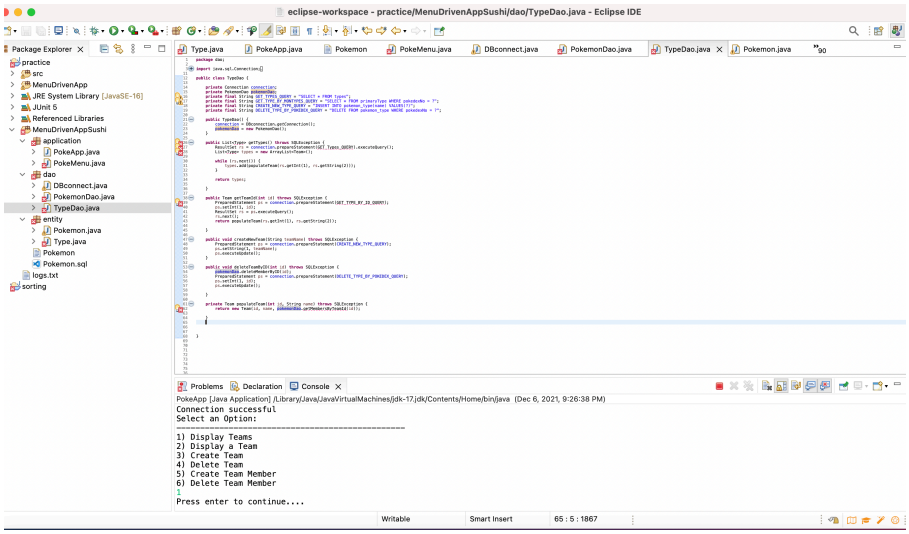
Caption



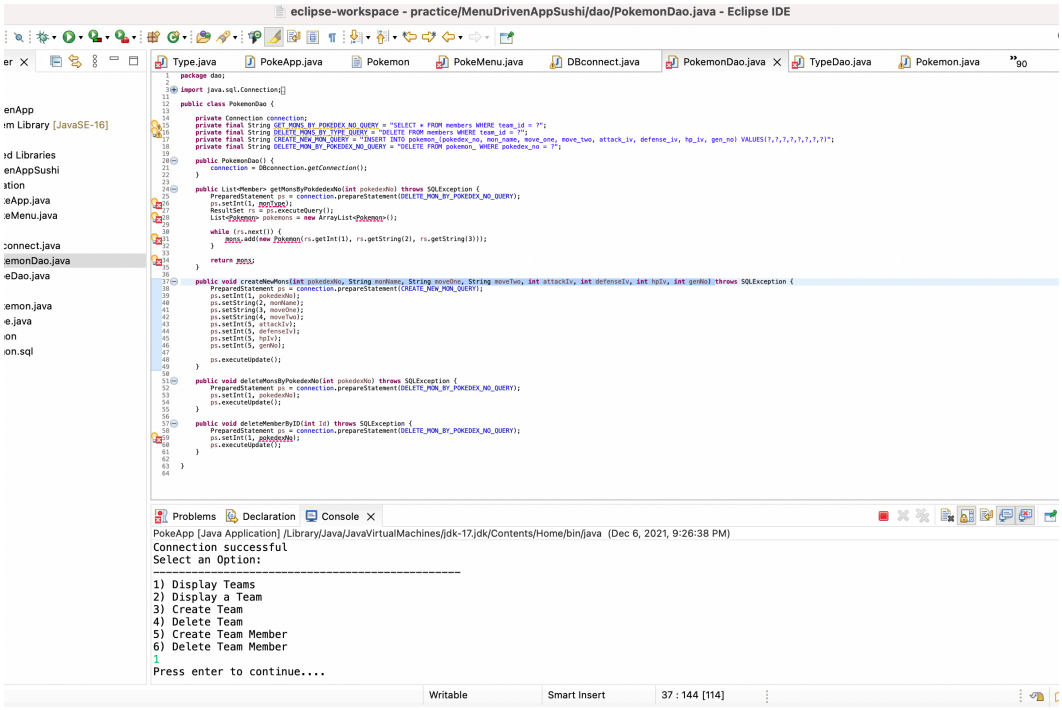
Caption

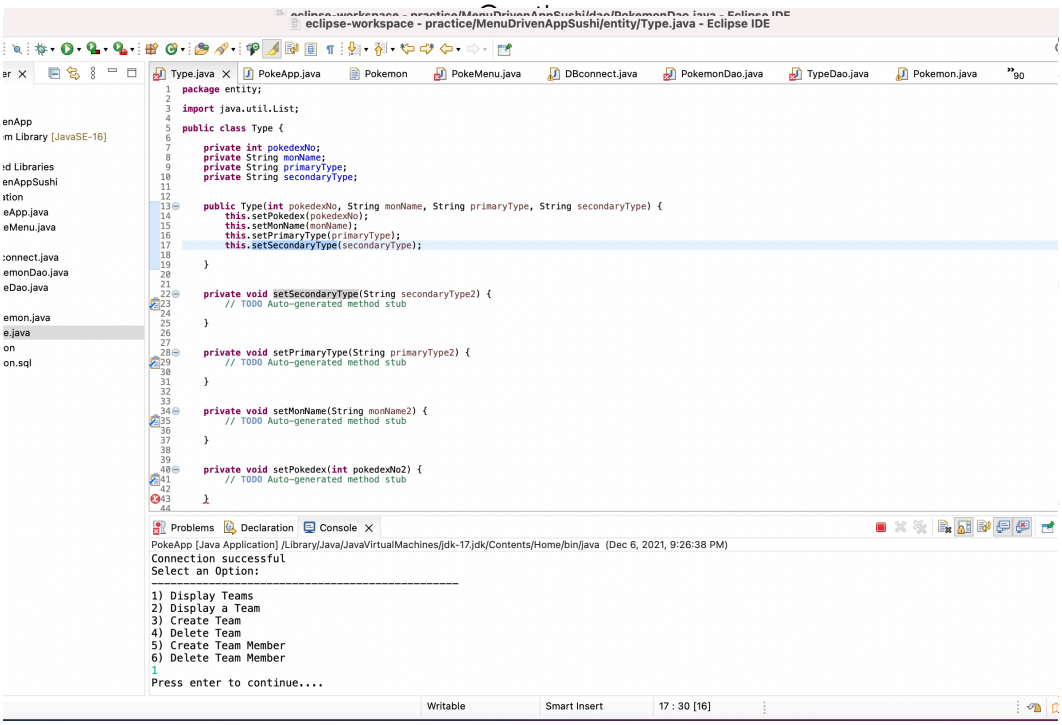


Caption

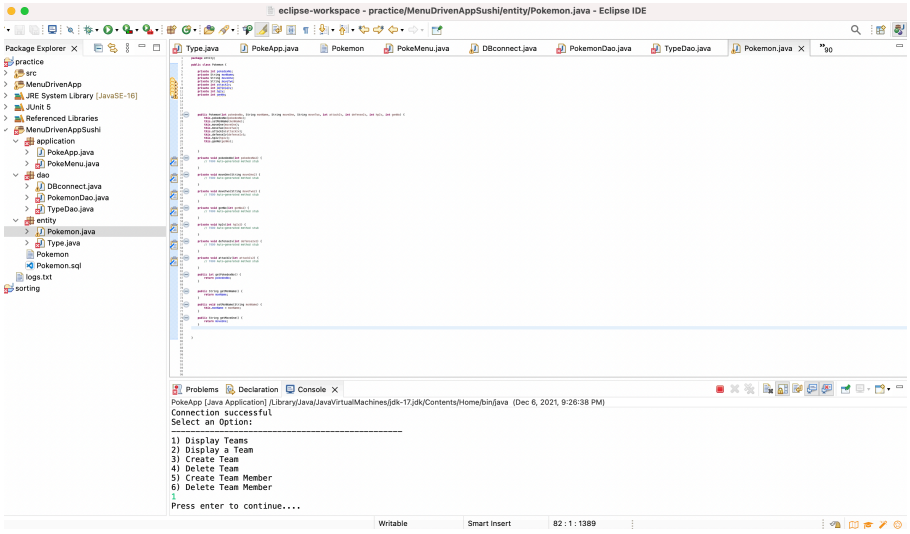

eclipse-workspace - practice/MenuDrivenAppSushi/dao/TypeDao.java - Eclipse IDE
Package Explorer
practice
src
MenuDrivenApp
JRE System Library [JavaSE-16]
JUnit 5
Referenced Libraries
MenuDrivenAppSushi
application
PokeApp.java
PokeMenu.java
dao
DBconnect.java
PokemonDao.java
TypeDao.java
entity
Pokemon.java
Type.java
Pokemon
Pokemon.sql
logs.txt
sorting
Type.java
PokeApp.java
Pokemon
PokeMenu.java
DBconnect.java
PokemonDao.java
TypeDao.java
Pokemon.java
Problems
Declaration
Console
PokeApp [Java Application] /Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home/bin/java (Dec 6, 2021, 9:26:38 PM)
Connection successful
Select an Option:
1) Display Teams
2) Display a Team
3) Create Team
4) Delete Team
5) Create Team Member
6) Delete Team Member
Press enter to continue....
Writable
Smart Insert
65 : 5 : 1867

Caption


eclipse-workspace - practice/MenuDrivenAppSushi/dao/PokemonDao.java - Eclipse IDE
Type.java
PokeApp.java
Pokemon
PokeMenu.java
DBconnect.java
PokemonDao.java
TypeDao.java
Pokemon.java
Problems
Declaration
Console
PokeApp [Java Application] /Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home/bin/java (Dec 6, 2021, 9:26:38 PM)
Connection successful
Select an Option:
1) Display Teams
2) Display a Team
3) Create Team
4) Delete Team
5) Create Team Member
6) Delete Team Member
1
Press enter to continue....
Writable
Smart Insert
37 : 144 [114]


eclipse-workspace - practice/MenuDrivenAppSushi/entity/Type.java - Eclipse IDE
Type.java
PokeApp.java
Pokemon
PokeMenu.java
DBconnect.java
PokemonDao.java
TypeDao.java
Pokemon.java
Problems
Declaration
Console
PokeApp [Java Application] /Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home/bin/java (Dec 6, 2021, 9:26:38 PM)
Connection successful
Select an Option:
1) Display Teams
2) Display a Team
3) Create Team
4) Delete Team
5) Create Team Member
6) Delete Team Member
1
Press enter to continue....
Writable
Smart Insert
17 : 30 [16]

Caption



Caption

URL to GitHub Repository: