Relational Databases with MySQL Week 5 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: 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:

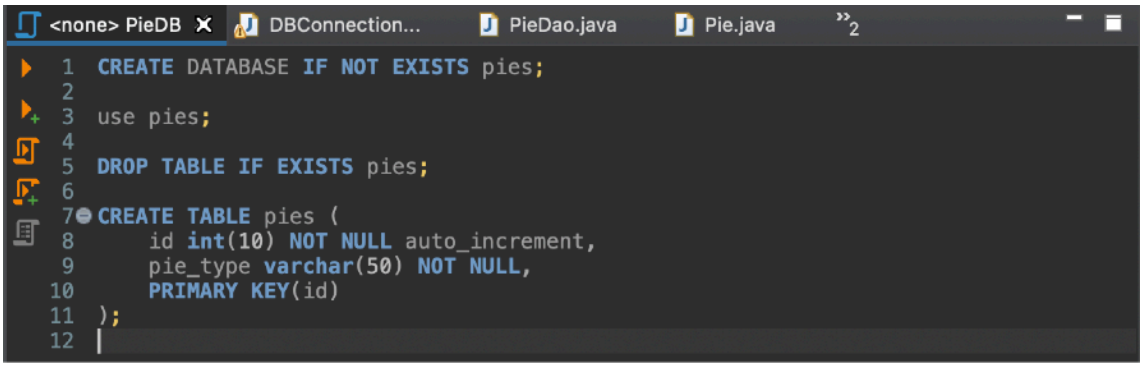
The application does not need to be as complex as the example in the video curriculum.

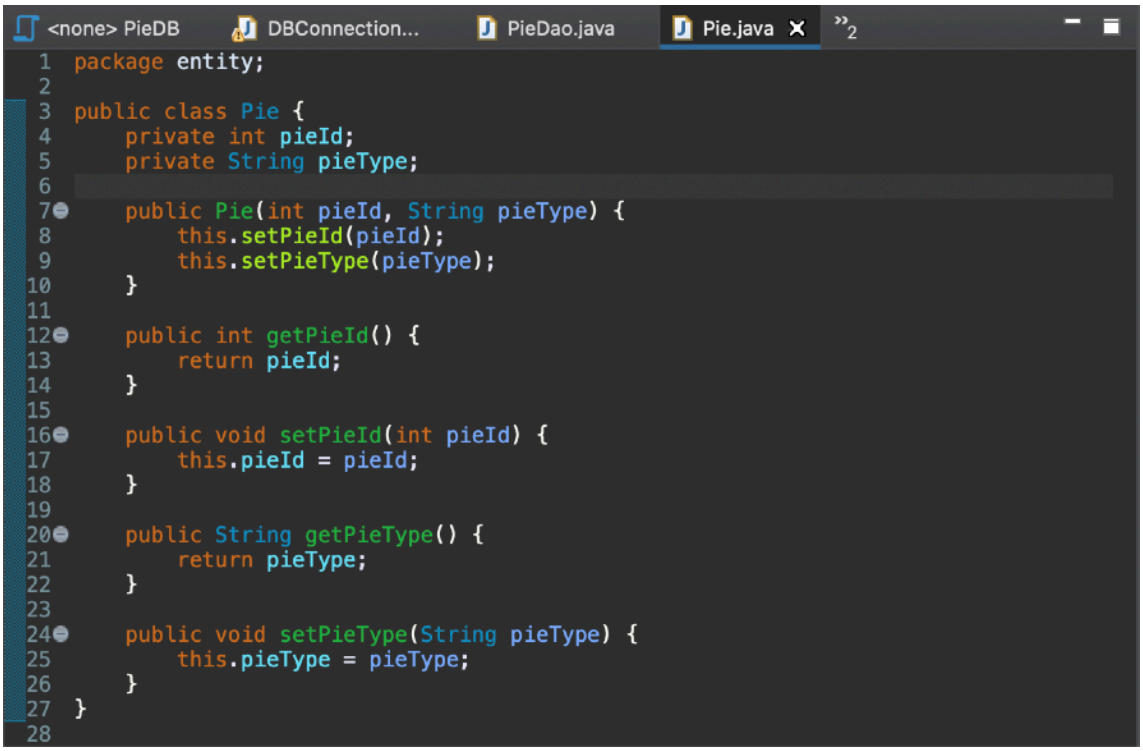
You need an option for each of the CRUD operations (Create, Read, Update, and Delete).

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

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

Screenshots of Code:



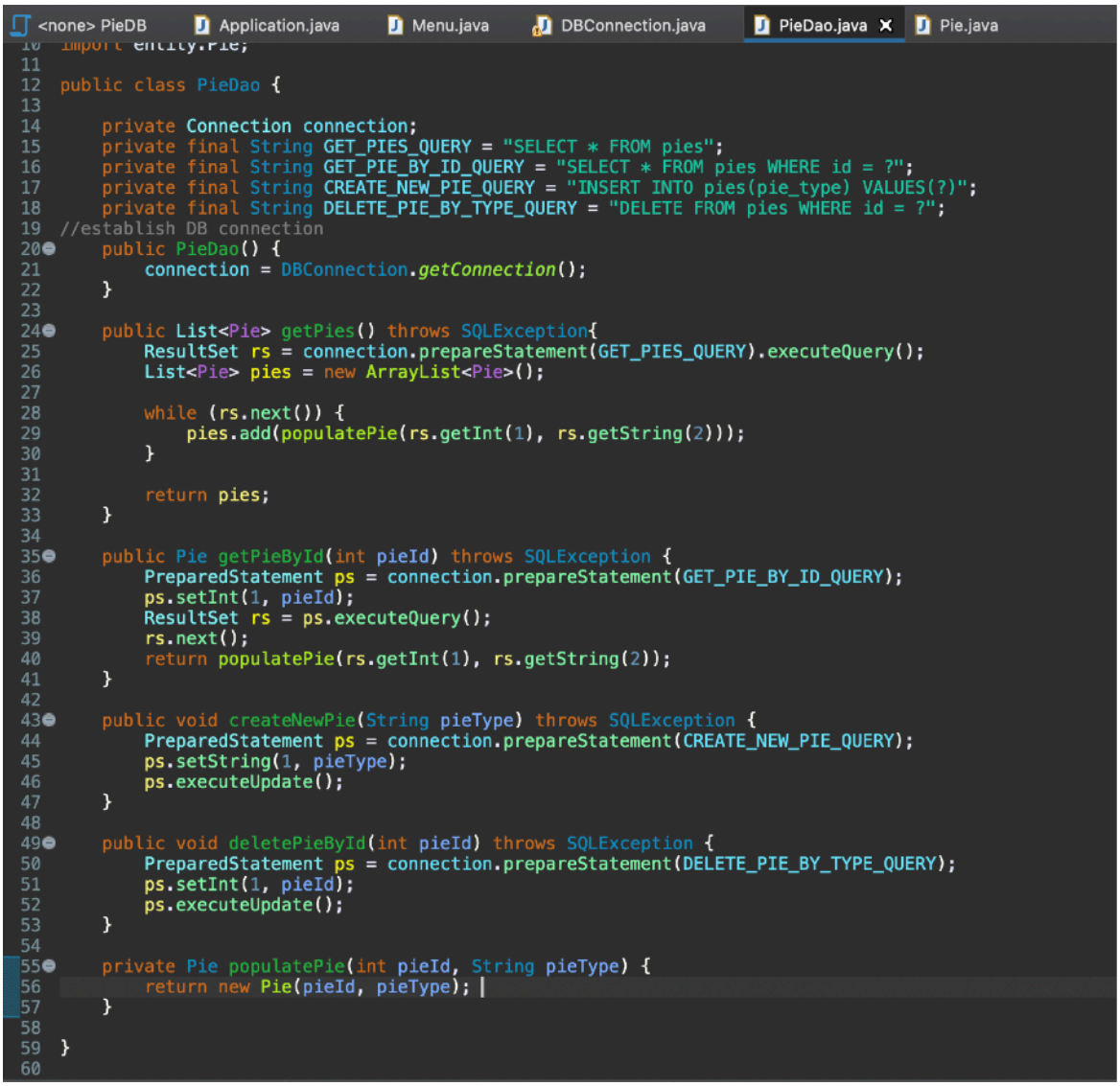


<none> PieDB | DBConnection... | PieDao.java | Pie.java

```java
package dao;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

//establish database connection
public class DBConnection {

    private final static String URL = "jdbc:mysql://localhost:3306/pies";
    private final static String USERNAME = System.getenv("username");
    private final static String PASSWORD = System.getenv("password");
    private static Connection connection;
    private static DBConnection instance;

    private DBConnection(Connection connection) {
        this.connection = connection;
    }

    public static Connection getConnection() {
        if (instance == null) {
            try {
                connection = DriverManager.getConnection(URL, USERNAME, PASSWORD)
                instance = new DBConnection(connection);
                System.out.println("Connection Successful!");
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        return DBConnection.connection;
    }

}
```

<none> PieDB | DBConnection... | PieDao.java | Pie.java

```java
package dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import entity.Pie;

public class PieDao {

    private Connection connection;
    private final String GET_PIES_QUERY = "SELECT * FROM pies";
    private final String GET_PIE_BY_ID_QUERY = "SELECT * FROM pies WHERE id = ?";
    private final String CREATE_NEW_PIE_QUERY = "INSERT INTO pies(pie_type) VALUE
    private final String DELETE_PIE_BY_TYPE_QUERY = "DELETE FROM pies WHERE pie_t
//establish DB connection
    public PieDao() {
        connection = DBConnection.getConnection();
    }

    public List<Pie> getPies() throws SQLException{
        ResultSet rs = connection.prepareStatement(GET_PIES_QUERY).executeQuery()
        List<Pie> pies = new ArrayList<Pie>();

        while (rs.next()) {
            pies.add(populatePie(rs.getInt(1), rs.getString(2)));
        }

        return pies;
    }

    public Pie getPieById(int pieId) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(GET_PIE_BY_ID_QUERY);
        ps.setInt(1, pieId);
        ResultSet rs = ps.executeQuery();
        rs.next();
        return populatePie(rs.getInt(1), rs.getString(2));
    }

```




```java
import entity.Pie;

public class PieDao {

    private Connection connection;
    private final String GET_PIES_QUERY = "SELECT * FROM pies";
    private final String GET_PIE_BY_ID_QUERY = "SELECT * FROM pies WHERE id = ?";
    private final String CREATE_NEW_PIE_QUERY = "INSERT INTO pies(pie_type) VALUES(?)";
    private final String DELETE_PIE_BY_TYPE_QUERY = "DELETE FROM pies WHERE id = ?";
//establish DB connection
    public PieDao() {
        connection = DBConnection.getConnection();
    }

    public List<Pie> getPies() throws SQLException{
        ResultSet rs = connection.prepareStatement(GET_PIES_QUERY).executeQuery();
        List<Pie> pies = new ArrayList<Pie>();

        while (rs.next()) {
            pies.add(populatePie(rs.getInt(1), rs.getString(2)));
        }

        return pies;
    }

    public Pie getPieById(int pieId) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(GET_PIE_BY_ID_QUERY);
        ps.setInt(1, pieId);
        ResultSet rs = ps.executeQuery();
        rs.next();
        return populatePie(rs.getInt(1), rs.getString(2));
    }

    public void createNewPie(String pieType) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(CREATE_NEW_PIE_QUERY);
        ps.setString(1, pieType);
        ps.executeUpdate();
    }

    public void deletePieById(int pieId) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(DELETE_PIE_BY_TYPE_QUERY);
        ps.setInt(1, pieId);
        ps.executeUpdate();
    }

    private Pie populatePie(int pieId, String pieType) {
        return new Pie(pieId, pieType);
    }

}

```

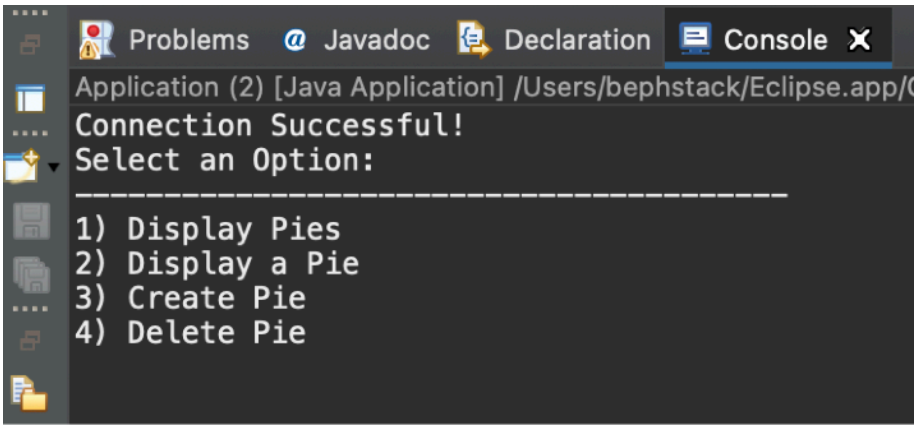

```java
package application;

public class Application {

    public static void main(String[] args) {
        Menu menu = new Menu();
        menu.start();
    }

}

```

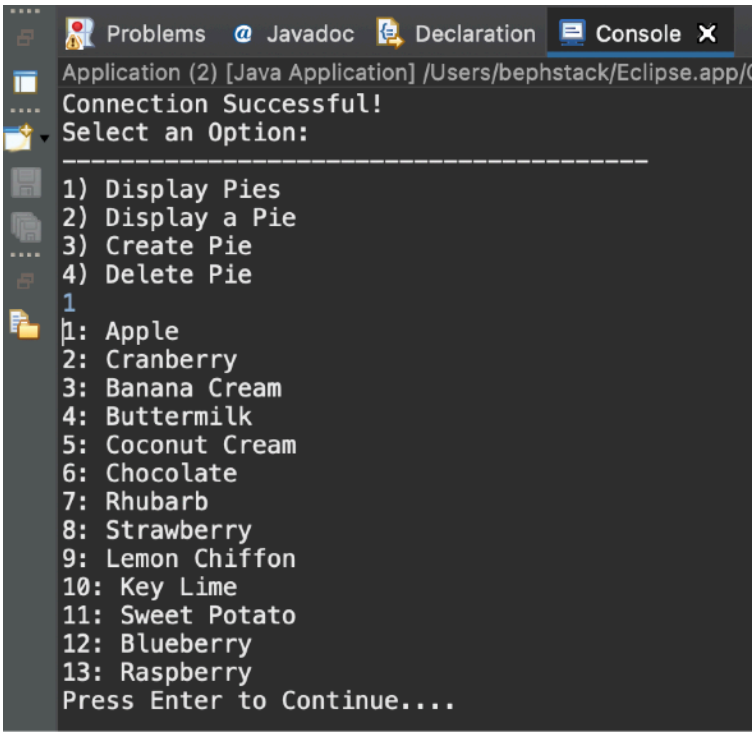
Application.... | *Menu.java | DBConnection... | *PieDao.java | Pie.java

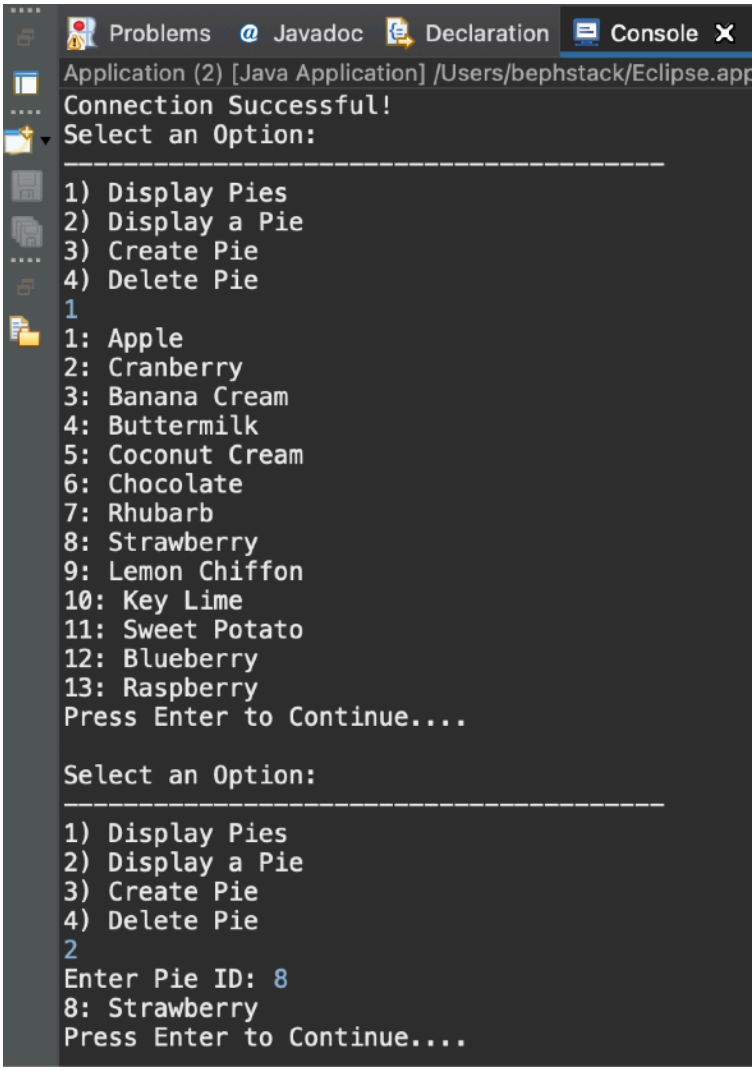
```java
package application;

import java.sql.SQLException;
import java.util.Arrays;
import java.util.List;
import java.util.Scanner;

import dao.PieDao;
import entity.Pie;

public class Menu {

    private PieDao pieDao = new PieDao();
    private Scanner scanner = new Scanner(System.in);
    private List<String> options = Arrays.asList("Display Pies",
            "Display a Pie",
            "Create Pie",
            "Delete Pie"
            );

    public void start() {
        String selection = "";

        do {
            printMenu();
            selection = scanner.nextLine();

            try {
                if(selection.equals("1")) {
                    displayPies();
                } else if (selection.equals("2")) {
                    displayPie();
                } else if (selection.equals("3")) {
                    createPie();
                } else if (selection.equals("4")) {
                    deletePie();
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }

            System.out.println("Press Enter to Continue....");
            scanner.nextLine();
        } while (!selection.equals("-1"));
    }

```

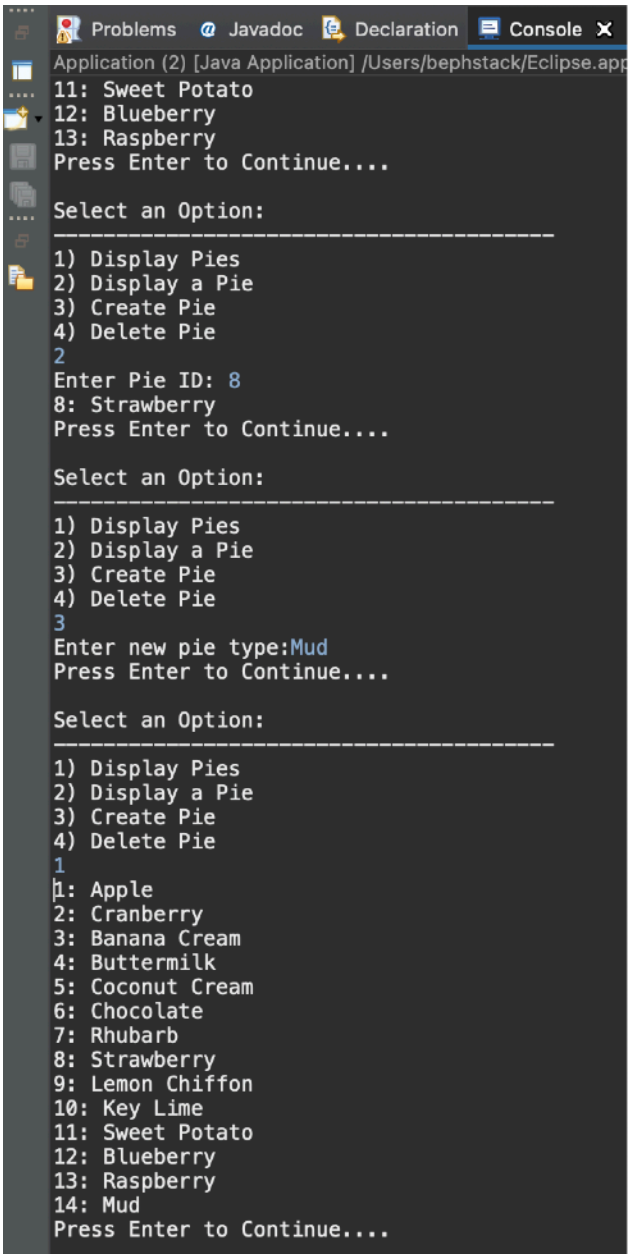
```
Application....  *Menu.java  DBConnection...  *PieDao.java  Pie.java
32                displayPie();
33            } else if (selection.equals("3")) {
34                createPie();
35            } else if (selection.equals("4")) {
36                deletePie();
37            }
38        } catch (SQLException e) {
39            e.printStackTrace();
40        }
41
42        System.out.println("Press Enter to Continue....");
43        scanner.nextLine();
44    } while (!selection.equals("-1"));
45 }
46
47 private void printMenu() {
48     System.out.println("Select an Option:\n----------------------------------------
49     for(int i = 0; i < options.size(); i++) {
50         System.out.println(i + 1 + ") " + options.get(i));
51     }
52 }
53
54 private void displayPies() throws SQLException {
55     List<Pie> pies = pieDao.getPies();
56     for (Pie pie : pies) {
57         System.out.println(pie.getPieId() + ": " + pie.getPieType());
58     }
59 }
60
61 private void displayPie() throws SQLException {
62     System.out.print("Enter Pie ID: ");
63     int pieId = Integer.parseInt(scanner.nextLine());
64     Pie pie = pieDao.getPieById(pieId);
65     System.out.println(pie.getPieId() + ": " + pie.getPieType());
66 }
67
68 private void createPie() throws SQLException {
69     System.out.print("Enter new pie type:");
70     String pieType = scanner.nextLine();
71     pieDao.createNewPie(pieType);
72 }
73
74 private void deletePie() throws SQLException {
75     System.out.print("Enter pie ID to delete:");
76     int id = Integer.parseInt(scanner.nextLine());
77     pieDao.deletePieById(id);
78 }
79
80 }
81
```

Screenshots of Running Application:




Problems
Javadoc
Declaration
Console
Application (2) [Java Application] /Users/bephstack/Eclipse.app/
Connection Successful!
Select an Option:
1) Display Pies
2) Display a Pie
3) Create Pie
4) Delete Pie
1
1: Apple
2: Cranberry
3: Banana Cream
4: Buttermilk
5: Coconut Cream
6: Chocolate
7: Rhubarb
8: Strawberry
9: Lemon Chiffon
10: Key Lime
11: Sweet Potato
12: Blueberry
13: Raspberry
Press Enter to Continue....


Problems
Javadoc
Declaration
Console
Application (2) [Java Application] /Users/bephstack/Eclipse.app
Connection Successful!
Select an Option:
1) Display Pies
2) Display a Pie
3) Create Pie
4) Delete Pie
1
1: Apple
2: Cranberry
3: Banana Cream
4: Buttermilk
5: Coconut Cream
6: Chocolate
7: Rhubarb
8: Strawberry
9: Lemon Chiffon
10: Key Lime
11: Sweet Potato
12: Blueberry
13: Raspberry
Press Enter to Continue....
Select an Option:
1) Display Pies
2) Display a Pie
3) Create Pie
4) Delete Pie
2
Enter Pie ID: 8
8: Strawberry
Press Enter to Continue....




```
Select an Option:
--------------------------------------
1) Display Pies
2) Display a Pie
3) Create Pie
4) Delete Pie
4
Enter pie ID to delete:14
Press Enter to Continue....

Select an Option:
--------------------------------------
1) Display Pies
2) Display a Pie
3) Create Pie
4) Delete Pie
1
1: Apple
2: Cranberry
3: Banana Cream
4: Buttermilk
5: Coconut Cream
6: Chocolate
7: Rhubarb
8: Strawberry
9: Lemon Chiffon
10: Key Lime
11: Sweet Potato
12: Blueberry
13: Raspberry
Press Enter to Continue....
```

URL to GitHub Repository: