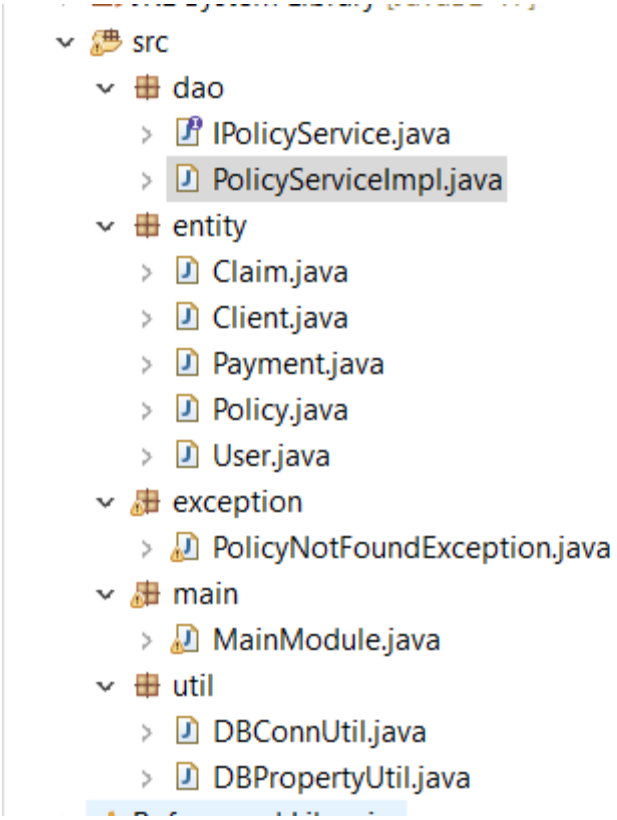
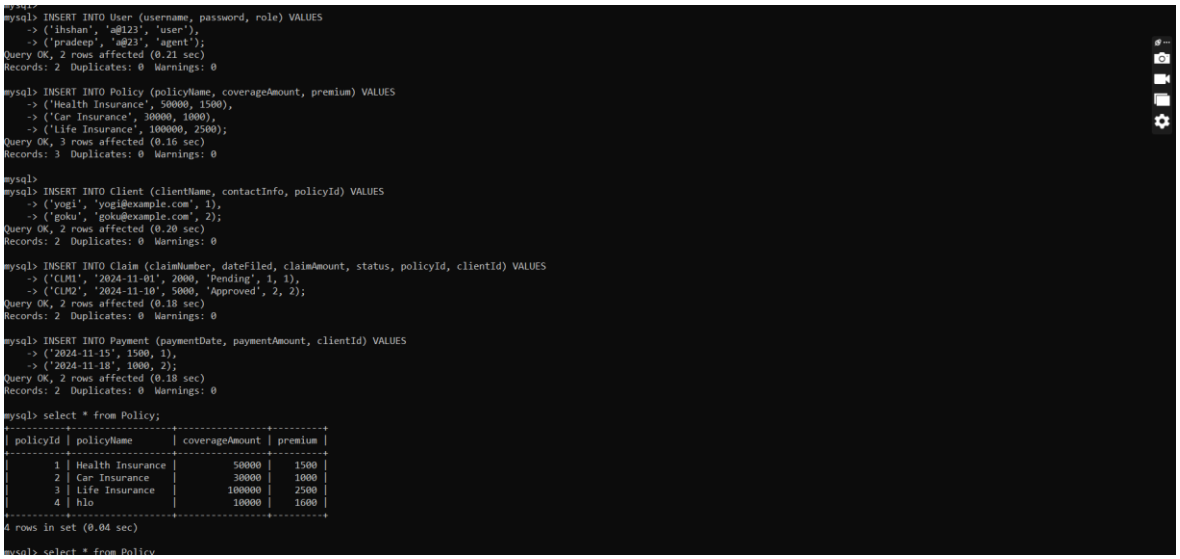
Java coding challenge-insurance

Project structure:



Sql schema for the tables:

Inserting values:



```
mysql> INSERT INTO User (username, password, role) VALUES
    -> ('ihshan', 'a@123', 'user'),
    -> ('pradeep', 'a@23', 'agent');
Query OK, 2 rows affected (0.21 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> INSERT INTO Policy (policyName, coverageAmount, premium) VALUES
    -> ('Health Insurance', 50000, 1500),
    -> ('Car Insurance', 30000, 1000),
    -> ('Life Insurance', 100000, 2500);
Query OK, 3 rows affected (0.16 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>
mysql> INSERT INTO Client (clientName, contactInfo, policyId) VALUES
    -> ('yogi', 'yogi@example.com', 1),
    -> ('goku', 'goku@example.com', 2);
Query OK, 2 rows affected (0.20 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> INSERT INTO Claim (claimNumber, dateFiled, claimAmount, status, policyId, clientId) VALUES
    -> ('CLM1', '2024-11-01', 2000, 'Pending', 1, 1),
    -> ('CLM2', '2024-11-10', 5000, 'Approved', 2, 2);
Query OK, 2 rows affected (0.18 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> INSERT INTO Payment (paymentDate, paymentAmount, clientId) VALUES
    -> ('2024-11-15', 1500, 1),
    -> ('2024-11-18', 1000, 2);
Query OK, 2 rows affected (0.18 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> select * from Policy;
+----------+------------------+----------------+---------+
| policyId | policyName       | coverageAmount | premium |
+----------+------------------+----------------+---------+
|        1 | Health Insurance |          50000 |    1500 |
|        2 | Car Insurance    |          30000 |    1000 |
|        3 | Life Insurance   |         100000 |    2500 |
|        4 | hlo              |          10000 |    1600 |
+----------+------------------+----------------+---------+
4 rows in set (0.04 sec)

mysql> select * from Policy
```

Entity package:

User.java

```java
package entity;

public class User {
    private int userId;
    private String username;
    private String password;
    private String role;

    public User(int userId, String username, String password, String role) {
        this.userId = userId;
        this.username = username;
        this.password = password;
        this.role = role;
    }

    public int getUserId() {
        return userId;
        }
    public void setUserId(int userId) {
        this.userId = userId;
        }

    public String getUsername() {
        return username;
        }
    public void setUsername(String username) {
        this.username = username;
        }

    public String getPassword() {
        return password;
        }
    public void setPassword(String password) {
        this.password = password;
        }

    public String getRole() {
        return role;
        }
```

Claim.java:

User.java | Claim.java

```java

import java.util.Date;

public class Claim {
    private int claimId;
    private String claimNumber;
    private Date dateFiled;
    private double claimAmount;
    private String status;
    private String policy;
    private Client client;

    public Claim(int claimId, String claimNumber, Date dateFiled, double claimAmount, String status, String policy, Client client) {
        this.claimId = claimId;
        this.claimNumber = claimNumber;
        this.dateFiled = dateFiled;
        this.claimAmount = claimAmount;
        this.status = status;
        this.policy = policy;
        this.client = client;
    }

    public int getClaimId() {
        return claimId;
        }
    public void setClaimId(int claimId) {
        this.claimId = claimId;
        }

    public String getClaimNumber() {
        return claimNumber;
        }
    public void setClaimNumber(String claimNumber) {
        this.claimNumber = claimNumber;
        }

    public Date getDateFiled() {
        return dateFiled;
        }
```

Client.java

User.java | Claim.java | Client.java

```java
package entity;

public class Client {
    private int clientId;
    private String clientName;
    private String contactInfo;
    private String policy;

    public Client(int clientId, String clientName, String contactInfo, String policy) {
        this.clientId = clientId;
        this.clientName = clientName;
        this.contactInfo = contactInfo;
        this.policy = policy;
    }

    public int getClientId() {
        return clientId;
        }
    public void setClientId(int clientId) {
        this.clientId = clientId;
        }

    public String getClientName() {
        return clientName;
        }
    public void setClientName(String clientName) {
        this.clientName = clientName;
        }

    public String getContactInfo() {
        return contactInfo;
        }
    public void setContactInfo(String contactInfo) {
        this.contactInfo = contactInfo;
        }

    public String getPolicy() {
        return policy;
        }
```

Policy.java:

```java
package entity;

public class Policy {
    private int policyId;
    private String policyName;
    private double coverageAmount;
    private double premium;

    public Policy(int policyId, String policyName, double coverageAmount, double premium) {
        this.policyId = policyId;
        this.policyName = policyName;
        this.coverageAmount = coverageAmount;
        this.premium = premium;
    }


    public int getPolicyId()
    {
        return policyId;
        }
    public void setPolicyId(int policyId) {
        this.policyId = policyId
        }

    public String getPolicyName(
        return policyName;
        }
    public void setPolicyName(String policyName) {
        this.policyName = policyName;
        }

    public double getCoverageAmount() {
        return coverageAmount;
        }
    public void setCoverageAmount(double coverageAmount) {
        this.coverageAmount = coverageAmount;
        }

    public double getPremium() {
```

int policyId - entity.Policy.setPolicyId (
int
)
Press 'F2' for focus

Util package:

```java
package util;


import java.sql.Connection;

public class DBConnUtil {
    private static Connection connection;

    public static Connection getConnection(String propertyFileName) {
        if (connection == null) {
            try {
                Properties props= DBPropertyUtil.loadProperties(propertyFileName);
                String driver = props.getProperty("driver");
                String url = props.getProperty("url");
                String username = props.getProperty("username");
                String password = props.getProperty("password");

                Class.forName(driver);
                connection = DriverManager.getConnection(url, username, password);
            } catch (ClassNotFoundException | SQLException e) {
                e.printStackTrace();
            }
        }
        return connection;
    }
}
```

DBpropertyutil.java

User.java | Claim.java | Client.java | Policy.java | DBConnUtil.java | DBPropertyUtil.java | db.properties

```java
package util;


import java.io.FileInputStream;

public class DBPropertyUtil {
    public static Properties loadProperties(String fileName) {
        Properties properties = new Properties();
        try (FileInputStream fis = new FileInputStream(fileName)) {
            properties.load(fis);
        } catch (IOException e) {
            e.printStackTrace();
        }
        return properties;
    }
}

```

Dao package

IPolicyService.java

User.java | Claim.java | Client.java | Policy.java | DBConnUtil.java | DBPropertyUtil.java | db.properties | IPolicyService.java | PolicyServiceImpl.ja

```java
package dao;

import entity.Policy;

public interface IPolicyService {
    boolean createPolicy(Policy policy);
    Policy getPolicy(int policyId);
    List<Policy> getAllPolicies();
    boolean updatePolicy(Policy policy);
    boolean deletePolicy(int policyId);
}

```

policyserviceImpl.java

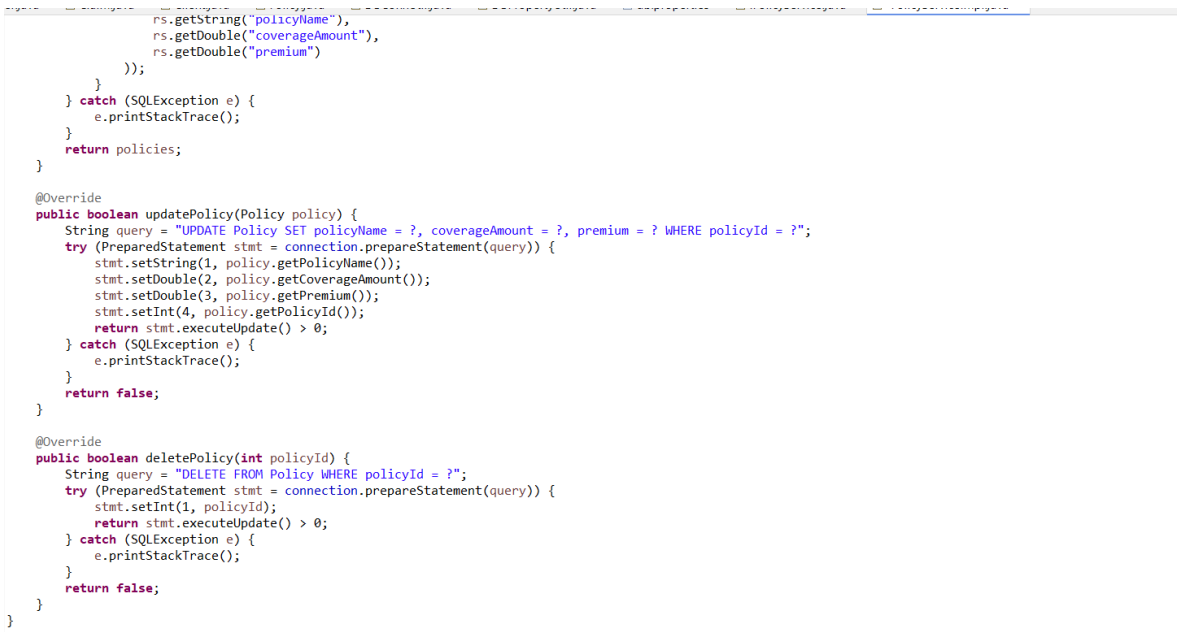
```java
import entity.Policy;

public class PolicyServiceImpl implements IPolicyService {
    private Connection connection;

    public PolicyServiceImpl() {
        this.connection = DBConnUtil.getConnection("db.properties");
    }

    @Override
    public boolean createPolicy(Policy policy) {
        String query = "INSERT INTO Policy (policyId, policyName, coverageAmount, premium) VALUES (?, ?, ?, ?)";
        try (PreparedStatement stmt = connection.prepareStatement(query)) {
            stmt.setInt(1, policy.getPolicyId());
            stmt.setString(2, policy.getPolicyName());
            stmt.setDouble(3, policy.getCoverageAmount());
            stmt.setDouble(4, policy.getPremium());
            return stmt.executeUpdate() > 0;
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return false;
    }

    @Override
    public Policy getPolicy(int policyId) {
        String query = "SELECT * FROM Policy WHERE policyId = ?";
        try (PreparedStatement stmt = connection.prepareStatement(query)) {
            stmt.setInt(1, policyId);
            ResultSet rs = stmt.executeQuery();
            if (rs.next()) {
                return new Policy(
                    rs.getInt("policyId"),
                    rs.getString("policyName"),
                    rs.getDouble("coverageAmount"),
                    rs.getDouble("premium")
                );
            }
        } catch (SQLException e) {
```



```java
                    rs.getString("policyName"),
                    rs.getDouble("coverageAmount"),
                    rs.getDouble("premium")
                ));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return policies;
    }

    @Override
    public boolean updatePolicy(Policy policy) {
        String query = "UPDATE Policy SET policyName = ?, coverageAmount = ?, premium = ? WHERE policyId = ?";
        try (PreparedStatement stmt = connection.prepareStatement(query)) {
            stmt.setString(1, policy.getPolicyName());
            stmt.setDouble(2, policy.getCoverageAmount());
            stmt.setDouble(3, policy.getPremium());
            stmt.setInt(4, policy.getPolicyId());
            return stmt.executeUpdate() > 0;
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return false;
    }

    @Override
    public boolean deletePolicy(int policyId) {
        String query = "DELETE FROM Policy WHERE policyId = ?";
        try (PreparedStatement stmt = connection.prepareStatement(query)) {
            stmt.setInt(1, policyId);
            return stmt.executeUpdate() > 0;
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return false;
    }
}
```

Main module

User.java | Claim.java | Client.java | Policy.java | DBConnUtil.java | DBPropert

```java
package exception;

public class PolicyNotFoundException extends Exception {
    public PolicyNotFoundException(String message) {
        super(message);
    }
}
```

```java
package main;

import dao.PolicyServiceImpl;

public class MainModule {
    public static void main(String[] args) {
        PolicyServiceImpl policyService = new PolicyServiceImpl();
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("\nInsurance Management System");
            System.out.println("1. Create Policy");
            System.out.println("2. Get Policy by ID");
            System.out.println("3. Get All Policies");
            System.out.println("4. Update Policy");
            System.out.println("5. Delete Policy");
            System.out.println("6. Exit");

            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.println("Enter Policy Details:");
                    System.out.print("Policy ID: ");
                    int policyId = scanner.nextInt();
                    scanner.nextLine();
                    System.out.print("Policy Name: ");
                    String policyName = scanner.nextLine();
                    System.out.print("Coverage Amount: ");
                    double coverageAmount = scanner.nextDouble();
                    System.out.print("Premium: ");
                    double premium = scanner.nextDouble();

                    Policy newPolicy = new Policy(policyId, policyName, coverageAmount, premium);
                    if (policyService.createPolicy(newPolicy)) {
                        System.out.println("Policy created successfully!");
                    } else {
                        System.out.println("Failed to create policy.");
```

exception

```
                break;

            case 4:
                System.out.println("Enter Updated Policy Details:");
                System.out.print("Policy ID: ");
                int updateId = scanner.nextInt();
                scanner.nextLine();
                System.out.print("Policy Name: ");
                String updateName = scanner.nextLine();
                System.out.print("Coverage Amount: ");
                double updateCoverage = scanner.nextDouble();
                System.out.print("Premium: ");
                double updatePremium = scanner.nextDouble();

                Policy updatePolicy = new Policy(updateId, updateName, updateCoverage, updatePremium);
                if (policyService.updatePolicy(updatePolicy)) {
                    System.out.println("Policy updated successfully!");
                } else {
                    System.out.println("Failed to update policy.");
                }
                break;

            case 5:
                System.out.print("Enter Policy ID to Delete: ");
                int deleteId = scanner.nextInt();
                if (policyService.deletePolicy(deleteId)) {
                    System.out.println("Policy deleted successfully!");
                } else {
                    System.out.println("Failed to delete policy.");
                }
                break;

            case 6:
                System.out.println("Exiting... Goodbye!");
                System.exit(0);

            default:
                System.out.println("Invalid choice. Please try again.");
        }
```

Result

```
MainModule [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe  (22 Nov 2024, 3:39:22 pm) [pid: 15932]
4. Update Policy
5. Delete Policy
6. Exit
Enter your choice: 3
All Policies:
Policy [policyId=1, policyName=Health Insurance, coverageAmount=50000.0, premium=1500.0]
Policy [policyId=2, policyName=222, coverageAmount=1000.0, premium=111.0]
Policy [policyId=3, policyName=Life Insurance, coverageAmount=100000.0, premium=2500.0]
Policy [policyId=4, policyName=hlo, coverageAmount=10000.0, premium=1600.0]

Insurance Management System
1. Create Policy
2. Get Policy by ID
3. Get All Policies
4. Update Policy
5. Delete Policy
6. Exit
Enter your choice: 2
Enter Policy ID: 3
Policy Details: Policy [policyId=3, policyName=Life Insurance, coverageAmount=100000.0, premium=2500.0]

Insurance Management System
1. Create Policy
2. Get Policy by ID
3. Get All Policies
```