Engenharia de Software - Padrões de Projeto

Pedro Eduardo Altran

Implementação do artigo da Oracle com DAO e J2EE

Projeto no GitHUB

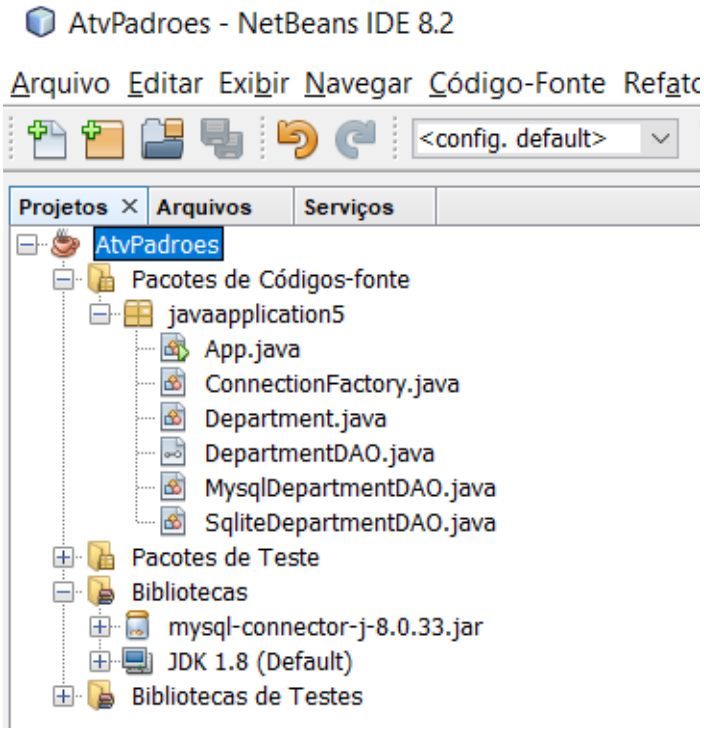
https://github.com/pedroaltran/Implementacao-artigo-Oracle

Criação das tabelas:

```
create table Departments
(
department_id integer not null primary key,
department_name varchar(255),
manager_id integer,
location_id varchar (255)
);


create table Employees (
employee_id integer not null primary key,
first_name varchar (255),
last_name varchar(255),
email varchar(255),
phone_number varchar(255),
hire_date varchar(10),
job_id varchar(255),
salary float,
comission_pct float,
manager_id integer,
department_id integer
);
```

Projeto Java:



App.java

```
package javaapplication5;

import java.sql.Connection;

/**
 *
 * @author Pedro Altran
 */
public class App {

    public static void main(String[] args) {

        //Conexão com MySQL e com SQLite
        ConnectionFactory connectionFactory = new ConnectionFactory();

        Connection mysqlConnection = connectionFactory.getMySqlConnection();
        if (mysqlConnection != null) {
            System.out.println("Conexão com MySQL estabelecida com sucesso!");
            testMysqlDepartmentDAO(mysqlConnection);
        } else {
            System.out.println("Falha ao estabelecer conexão com o banco de dados
MySQL.");
        }

        Connection sqliteConnection = connectionFactory.getSqliteConnection();
        if (sqliteConnection != null) {
            System.out.println("Conexão com SQLite estabelecida com sucesso!");
            testSqliteDepartmentDAO(sqliteConnection);
        } else {
            System.out.println("Falha ao estabelecer conexão com o banco de dados
SQLite.");
        }
    }

    //Testando os métodos no MySQL
    private static void testMysqlDepartmentDAO(Connection connection) {

        Department department = new Department();
        department.setDepartment_id(1);
        department.setDepartment_name("Teste MySQL");
        department.setManager_id(123);
        department.setLocation_id("12345");


        MysqlDepartmentDAO departmentDAO = new MysqlDepartmentDAO(connection);
```

```
        int rowsAffected = departmentDAO.insertDepartment(department);
        System.out.println("Linhas afetadas (insertDepartment): " + rowsAffected);

        Department retrievedDepartment = departmentDAO.findDepartment(1);
        System.out.println("Department encontrado (findDepartment): " +
retrievedDepartment);

        department.setDepartment_name("Departamento Atualizado");
        boolean updateSuccess = departmentDAO.updateDepartment(department);
        System.out.println("Departamento atualizado com sucesso?
(updateDepartment): " + updateSuccess);

        boolean deleteSuccess = departmentDAO.deleteDepartment(1);
        System.out.println("Departamento excluído com sucesso? (deleteDepartment):
" + deleteSuccess);
    }

    //Testando os métodos no SQLite
    private static void testSqliteDepartmentDAO(Connection connection) {

        Department department = new Department();
        department.setDepartment_id(2);
        department.setDepartment_name("Teste SQLite");
        department.setManager_id(456);
        department.setLocation_id("54321");

        SqliteDepartmentDAO departmentDAO = new SqliteDepartmentDAO(connection);

        int rowsAffected = departmentDAO.insertDepartment(department);
        System.out.println("Linhas afetadas : " + rowsAffected);

        Department retrievedDepartment = departmentDAO.findDepartment(2);
        System.out.println("Department encontrado: " + retrievedDepartment);

        department.setDepartment_name("Departamento Atualizado");
        boolean updateSuccess = departmentDAO.updateDepartment(department);
        System.out.println("Departamento atualizado com sucesso? " +
updateSuccess);

        boolean deleteSuccess = departmentDAO.deleteDepartment(2);
        System.out.println("Departamento excluído com sucesso? " + deleteSuccess);
    }
}
```

ConnectionFactory.java

```
package javaapplication5;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;


public class ConnectionFactory {
    public Connection getMySqlConnection(){
        try {
            return
DriverManager.getConnection("jdbc:mysql://localhost:3306/artigooracle","root","");
        }
        catch (SQLException e){
            System.out.println("Erro ao conectar ao banco de dados: " +
e.getMessage());
        }
        return null;
    }

     public Connection getSqliteConnection() {
        try {
            return DriverManager.getConnection("jdbc:sqlite:C:/Users/Program
Files/Docs/SQLite/artigooracle.sqlite");
        } catch (SQLException e) {
            System.out.println("Erro ao conectar ao banco de dados SQLite: " +
e.getMessage());
        }
        return null;
    }
}
```

Department.java

```
package javaapplication5;

public class Department implements java.io.Serializable{
    private int department_id;
    private String department_name;
    private int manager_id;
    private String location_id;

    public int getDepartment_id() {
        return department_id;
    }

    public void setDepartment_id(int department_id) {
        this.department_id = department_id;
    }

    public String getDepartment_name() {
        return department_name;
    }

    public void setDepartment_name(String department_name) {
        this.department_name = department_name;
    }

    public int getManager_id() {
        return manager_id;
    }

    public void setManager_id(int manager_id) {
        this.manager_id = manager_id;
    }

    public String getLocation_id() {
        return location_id;
    }

    public void setLocation_id(String location_id) {
        this.location_id = location_id;
    }
}
```

DepartmentDAO.java

```
package javaapplication5;
import javax.sql.rowset.JdbcRowSet;

public interface DepartmentDAO {
    public int insertDepartment(Department department);
    public boolean deleteDepartment(int departmentId);
    public Department findDepartment(int departmentId);
    public JdbcRowSet selectDepartmentsRS();
    public boolean updateDepartment(Department department);
}
```

MysqlDepartmentDAO.java

```
package javaapplication5;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.sql.rowset.JdbcRowSet;
import javax.sql.rowset.RowSetFactory;
import javax.sql.rowset.RowSetProvider;


public class MysqlDepartmentDAO implements DepartmentDAO {
        private final Connection connection;

    public MysqlDepartmentDAO(Connection connection) {
        this.connection = connection;
    }

    @Override
    public int insertDepartment(Department department) {
        String sql = "INSERT INTO Departments (department_id, department_name,
manager_id, location_id) VALUES (?, ?, ?, ?)";
        try (PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setInt(1, department.getDepartment_id());
            statement.setString(2, department.getDepartment_name());
            statement.setInt(3, department.getManager_id());
            statement.setString(4, department.getLocation_id());
            return statement.executeUpdate();
        } catch (SQLException e) {
            System.out.println("Erro ao conectar ao banco de dados: " +
e.getMessage());
        }
        return 0;
    }

    @Override
    public boolean deleteDepartment(int departmentId) {
        String sql = "DELETE FROM Departments WHERE department_id = ?";
        try (PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setInt(1, departmentId);
            int rowsAffected = statement.executeUpdate();
            return rowsAffected > 0;
        } catch (SQLException e) {
            System.out.println("Erro ao conectar ao banco de dados: " +
e.getMessage());
        }
        return false;
```

```
    }

    @Override
    public Department findDepartment(int departmentId) {
            String sql = "SELECT * FROM Departments WHERE department_id = ?";
            try (PreparedStatement statement = connection.prepareStatement(sql)) {
                statement.setInt(1, departmentId);
                ResultSet resultSet = statement.executeQuery();
                if (resultSet.next()) {
                    Department department = new Department();
                    department.setDepartment_id(resultSet.getInt("department_id"));

department.setDepartment_name(resultSet.getString("department_name"));
                    department.setManager_id(resultSet.getInt("manager_id"));
                    department.setLocation_id(resultSet.getString("location_id"));
                    return department;
                }
            } catch (SQLException e) {
                System.out.println("Erro ao conectar ao banco de dados: " +
e.getMessage());
            }
            return null;
        }

    @Override
    public JdbcRowSet selectDepartmentsRS() {
        String sql = "SELECT * FROM Departments";
        try (PreparedStatement statement = connection.prepareStatement(sql);
             ResultSet resultSet = statement.executeQuery()) {
            RowSetFactory rowSetFactory = RowSetProvider.newFactory();
            JdbcRowSet rowSet = rowSetFactory.createJdbcRowSet();
            rowSet.setCommand(sql);
            rowSet.execute();
            return rowSet;
        } catch (SQLException e) {
            System.out.println("Erro ao conectar ao banco de dados: " +
e.getMessage());
        }
        return null;
    }

    @Override
    public boolean updateDepartment(Department department) {
        String sql = "UPDATE Departments SET department_name = ?, manager_id = ?,
location_id = ? WHERE department_id = ?";
        try (PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setString(1, department.getDepartment_name());
            statement.setInt(2, department.getManager_id());
            statement.setString(3, department.getLocation_id());
            statement.setInt(4, department.getDepartment_id());
            int rowsAffected = statement.executeUpdate();
```

```
            return rowsAffected > 0;
        } catch (SQLException e) {
            System.out.println("Erro ao conectar ao banco de dados: " +
e.getMessage());
        }
        return false;
    }
}
```

SqliteDepartmentDAO.java

```
package javaapplication5;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.sql.rowset.JdbcRowSet;
import javax.sql.rowset.RowSetFactory;
import javax.sql.rowset.RowSetProvider;

public class SqliteDepartmentDAO implements DepartmentDAO {
    private Connection connection;

    public SqliteDepartmentDAO(Connection connection) {
        this.connection = connection;
    }

    @Override
    public int insertDepartment(Department department) {
        String sql = "INSERT INTO Departments (department_id, department_name,
manager_id, location_id) VALUES (?, ?, ?, ?)";
        try (PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setInt(1, department.getDepartment_id());
            statement.setString(2, department.getDepartment_name());
            statement.setInt(3, department.getManager_id());
            statement.setString(4, department.getLocation_id());
            return statement.executeUpdate();
        } catch (SQLException e) {
            System.out.println("Erro ao conectar ao banco de dados: " +
e.getMessage());
        }
        return 0;
    }

    @Override
    public boolean deleteDepartment(int departmentId) {
        String sql = "DELETE FROM Departments WHERE department_id = ?";
        try (PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setInt(1, departmentId);
            int rowsAffected = statement.executeUpdate();
            return rowsAffected > 0;
        } catch (SQLException e) {
            System.out.println("Erro ao conectar ao banco de dados: " +
e.getMessage());
        }
        return false;
    }
```

```
    @Override
    public Department findDepartment(int departmentId) {
        String sql = "SELECT * FROM Departments WHERE department_id = ?";
        try (PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setInt(1, departmentId);
            ResultSet resultSet = statement.executeQuery();
            if (resultSet.next()) {
                Department department = new Department();
                department.setDepartment_id(resultSet.getInt("department_id"));

department.setDepartment_name(resultSet.getString("department_name"));
                department.setManager_id(resultSet.getInt("manager_id"));
                department.setLocation_id(resultSet.getString("location_id"));
                return department;
            }
        } catch (SQLException e) {
            System.out.println("Erro ao conectar ao banco de dados: " +
e.getMessage());
        }
        return null;
    }

    @Override
    public JdbcRowSet selectDepartmentsRS() {
        String sql = "SELECT * FROM Departments";
        try (PreparedStatement statement = connection.prepareStatement(sql);
             ResultSet resultSet = statement.executeQuery()) {
            RowSetFactory rowSetFactory = RowSetProvider.newFactory();
            JdbcRowSet rowSet = rowSetFactory.createJdbcRowSet();
            rowSet.setCommand(sql);
            rowSet.execute();
            return rowSet;
        } catch (SQLException e) {
            System.out.println("Erro ao conectar ao banco de dados: " +
e.getMessage());
        }
        return null;
    }

    @Override
    public boolean updateDepartment(Department department) {
        String sql = "UPDATE Departments SET department_name = ?, manager_id = ?,
location_id = ? WHERE department_id = ?";
        try (PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setString(1, department.getDepartment_name());
            statement.setInt(2, department.getManager_id());
            statement.setString(3, department.getLocation_id());
            statement.setInt(4, department.getDepartment_id());
            int rowsAffected = statement.executeUpdate();
            return rowsAffected > 0;
```

```
        } catch (SQLException e) {
            System.out.println("Erro ao conectar ao banco de dados: " +
e.getMessage());
        }
        return false;
    }
}
```