**package** br.edu.unicid.util;

**import** java.sql.\*;

**public** **class** ConnectionFactory {

**public** **static** Connection getConnection() **throws** Exception {

**try** {

// indica qual é o banco de dados que estou utilizando e seu driver

Class.*forName*("com.mysql.jdbc.Driver");

// estabelece a conexao e retorna uma conexao

String url = "jdbc:mysql://localhost:3306/DBAlunoCC";

String login = "root";

String senha = "";

return DriverManager.getConnection(url,login,senha);

} catch (Exception e) {

throw new Exception(e.getMessage());

}

}

// fecha uma conexão de três formas: conn, stmt, rs

**public** **static** **void** closeConnection(Connection conn, Statement stmt,

ResultSet rs) **throws** Exception {

*close*(conn, stmt, rs);

}

**public** **static** **void** closeConnection(Connection conn, Statement stmt)

**throws** Exception {

*close*(conn, stmt, **null**);

}

**public** **static** **void** closeConnection(Connection conn) **throws** Exception {

*close*(conn, **null**, **null**);

}

**private** **static** **void** close(Connection conn, Statement stmt, ResultSet rs)

**throws** Exception {

**try** {

**if** (rs != **null**)

rs.close();

**if** (stmt != **null**)

stmt.close();

**if** (conn != **null**)

conn.close();

} **catch** (Exception e) {

**throw** **new** Exception(e.getMessage());

}

}

}

**Classe Aluno.java**

**package** br.edu.unicid.bean;

**import** java.util.Date;

**public** **class** Aluno {

// padrão JavaBean ou POJO – Classe com getters/setters, mais métodos construtores

**private** **int** caAluno;

**private** String nomeAluno;

**private** String emailAluno;

**private** Date dtaNasc;

**private** **int** idadeAluno;

**private** String endAluno;

**public** Aluno(**int** caAluno, String nomeAluno, String emailAluno,

Date dtaNasc, **int** idadeAluno, String endAluno) {

**this**.caAluno = caAluno;

**this**.nomeAluno = nomeAluno;

**this**.emailAluno = emailAluno;

**this**.dtaNasc = dtaNasc;

**this**.idadeAluno = idadeAluno;

**this**.endAluno = endAluno;

}

**public** Aluno() {

}

**public** **int** getCaAluno() {

**return** caAluno;

}

**public** **void** setCaAluno(**int** caAluno) {

**this**.caAluno = caAluno;

}

**public** String getNomeAluno() {

**return** nomeAluno;

}

**public** **void** setNomeAluno(String nomeAluno) {

**this**.nomeAluno = nomeAluno;

}

**public** String getEmailAluno() {

**return** emailAluno;

}

**public** **void** setEmailAluno(String emailAluno) {

**this**.emailAluno = emailAluno;

}

**public** Date getDtaNasc() {

**return** dtaNasc;

}

**public** **void** setDtaNasc(Date dtaNasc) {

**this**.dtaNasc = dtaNasc;

}

**public** **int** getIdadeAluno() {

**return** idadeAluno;

}

**public** **void** setIdadeAluno(**int** idadeAluno) {

**this**.idadeAluno = idadeAluno;

}

**public** String getEndAluno() {

**return** endAluno;

}

**public** **void** setEndAluno(String endAluno) {

**this**.endAluno = endAluno;

}

}

**Classe AlunoDAO.java**

**package** br.edu.unicid.dao;

**import** java.sql.\*;

**import** java.util.ArrayList;

**import** java.util.List;

**import** br.edu.unicid.bean.Aluno;

**import** br.edu.unicid.util.ConnectionFactory;

**public** **class** AlunoDAO {

**private** Connection conn;

**private** PreparedStatement ps;

**private** ResultSet rs;

**private** Aluno aluno;

**public** AlunoDAO() **throws** Exception {

// chama a classe ConnectionFactory e estabele uma conexão

**try** {

**this**.conn = ConnectionFactory.*getConnection*();

} **catch** (Exception e) {

**throw** **new** Exception("erro: \n" + e.getMessage());

}

}

// método de salvar

**public** **void** salvar(Aluno aluno) **throws** Exception {

**if** (aluno == **null**)

**throw** **new** Exception("O valor passado nao pode ser nulo");

**try** {

String SQL = "INSERT INTO tbAluno (caAluno, nomeAluno, emailAluno, dtaNasc, endAluno,idadeAluno) values (?, ?, ?, ?, ?, ?)";

conn = **this**.conn;

ps = conn.prepareStatement(SQL);

ps.setInt(1, aluno.getCaAluno());

ps.setString(2, aluno.getNomeAluno());

ps.setString(3, aluno.getEmailAluno());

ps.setString(4, aluno.getDtaNasc());

ps.setString(5, aluno.getEndAluno());

ps.setInt(6, aluno.getIdadeAluno());

ps.executeUpdate();

} **catch** (SQLException sqle) {

**throw** **new** Exception("Erro ao inserir dados " + sqle);

} **finally** {

ConnectionFactory.*closeConnection*(conn, ps);

}

}

// método de atualizar

**public** **void** atualizar(Aluno aluno) **throws** Exception {

**if** (aluno == **null**)

**throw** **new** Exception("O valor passado nao pode ser nulo");

**try** {

String SQL = "UPDATE tbAluno set nomeAluno=?, emailAluno=?, dtaNasc=?, endAluno=?,idadeAluno=? WHERE caAluno = ?";

conn = **this**.conn;

ps = conn.prepareStatement(SQL);

ps.setString(1, aluno.getNomeAluno());

ps.setString(2, aluno.getEmailAluno());

ps.setDate(3, aluno.getDtaNasc());

ps.setString(4, aluno.getEndAluno());

ps.setInt(5, aluno.getIdadeAluno());

ps.setInt(6, aluno.getCaAluno());

ps.executeUpdate();

} **catch** (SQLException sqle) {

**throw** **new** Exception("Erro ao alterar dados " + sqle);

} **finally** {

ConnectionFactory.*closeConnection*(conn, ps);

}

}

// método de excluir

**public** **void** excluir(Aluno aluno) **throws** Exception {

**if** (aluno == **null**)

**throw** **new** Exception("O valor passado nao pode ser nulo");

**try** {

String SQL = "DELETE FROM tbAluno WHERE caAluno = ?";

conn = **this**.conn;

ps = conn.prepareStatement(SQL);

ps.setInt(1, aluno.getCaAluno());

ps.executeUpdate();

} **catch** (SQLException sqle) {

**throw** **new** Exception("Erro ao excluir dados " + sqle);

} **finally** {

ConnectionFactory.*closeConnection*(conn, ps);

}

}

// Procurar Aluno

**public** Aluno procurarAluno(**int** caAluno) **throws** Exception {

**try** {

String SQL = "SELECT \* FROM tbAluno WHERE caAluno=?";

conn = **this**.conn;

ps = conn.prepareStatement(SQL);

ps.setInt(1, caAluno);

rs = ps.executeQuery();

**if** (rs.next()) {

**int** ca = rs.getInt(1);

String nome = rs.getString(2);

String email = rs.getString(3);

String nascimento = rs.getString(4);

**int** idade = rs.getInt(5);

String endereco = rs.getString(6);

aluno = **new** Aluno(ca, nome, email, nascimento, idade, endereco);

}

**return** aluno;

} **catch** (SQLException sqle) {

**throw** **new** Exception(sqle);

} **finally** {

ConnectionFactory.*closeConnection*(conn, ps, rs);

}

}

// Listar todos os alunos

**public** List todosAlunos() **throws** Exception {

**try** {

conn = **this**.conn;

ps = conn.prepareStatement("SELECT \* FROM tbAluno");

rs = ps.executeQuery();

List<Aluno> list = **new** ArrayList<Aluno>();

**while** (rs.next()) {

**int** ca = rs.getInt(1);

String nome = rs.getString(2);

String email = rs.getString(3);

String nascimento = rs.getString(4);

**int** idade = rs.getInt(5);

String endereco = rs.getString(6);

list.add(**new** Aluno(ca, nome, email, nascimento, idade, endereco));

}

**return** list;

} **catch** (SQLException sqle) {

**throw** **new** Exception(sqle);

} **finally** {

ConnectionFactory.*closeConnection*(conn, ps, rs);

}

}

}