

POO - PROGRAMAÇÃO ORIENTADA A OBJETOS

Conectividade com Banco de Dados

Rodrigo R Silva

Instituto Federal de Educação, Ciência e Tecnologia Sul-Rio-Grandense
Campus Bagé

Nesta Aula Veremos...

1 Introdução

2 JDBC

3 Métodos de Connection

Introdução

Conceito

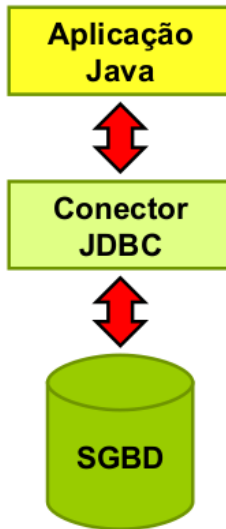
- Java apresenta uma biblioteca de classes e interfaces para conectividade de aplicações com Sistemas Gerenciadores de Banco de Dados (SGBD).
- **JDBC** – Java Database Connectivity.
- A JDBC provê um conjunto de recursos que permitem a uma aplicação cliente realizar conexão e manipulação de dados mantidos por um SGBD.
- A JDBC foi concebida para manipulação de bancos de dados relacionais, tecnologia dominante no mercado atual.

JDBC

JDBC

- As classes e interfaces da JDBC estão contidas no pacote **java.sql** o qual deve ser importado por aplicações que manipulam SGBD em Java.
- Uma aplicação Java realiza conexão com um SGBD através do conceito de **conector** (driver).
- Um conector é um pacote java (jar) que realiza a comunicação de uma aplicação com um SGBD.
- Cada SGBD deve fornecer seu conector para que o mesmo seja utilizado por uma aplicação Java.

Conectividade Java com SGBD



Registrando o conector

- Uma conexão Java com SGBD é gerenciada por uma classe denominada DriverManager.
- DriverManager é encarregada de gerenciar todos os conectores de banco de dados de uma aplicação.
- Para isso, a aplicação Java deverá registrar o conector de um banco de dados específico para que possa estabelecer a conexão.
- Somente após o registro do conector é que a aplicação Java poderá solicitar a abertura de uma conexão com um SGBD.

DriverManager

- DriverManager é a classe responsável por:
- Registrar o connector e abrir uma conexão.
- Registrando um conector:
- `DriverManager.registerDriver(new org.postgresql.Driver());`
- Método utilizado para registro do conector. O exemplo ilustra o registro do conector PostgreSQL.

Estabelecendo uma conexão

- Após o registro do conector no DriverManager já podemos estabelecer uma conexão com o SGBD.
- A conexão é estabelecida pelo método `getConnection()`, veja o exemplo abaixo:
- `Connection con = DriverManager.getConnection("jdbc:postgresql://localhost/bd", "postgres", "postgres");`
- No exemplo é apresentada uma conexão com PostgreSQL.
- O método `getConnection()` recebe como argumento a string de conexão, usuário e senha do banco. O método retorna um objeto do tipo `Connection`.

Connection

- Connection é uma interface definida na API JDBC que fornece métodos para:
- Controle de transações (commit e rollback).
- Criação e manipulação de instruções SQL.
- A instância de uma conexão é obtida através do método de classe getConnection() provido por DriverManager.

Métodos de Connection

Métodos

- void setAutoCommit(boolean autoCommit)
 - 1 Determina se os dados serão gravados no SGBD no momento da execução de uma instrução SQL de atualização de dados.
 - 2 Uso: con.setAutoCommit(true);
- void commit()
 - 1 Esvazia o buffer enviando efetivamente os dados para o SGBD, finalizando a transação.
 - 2 Uso: con.commit();

Métodos

- `void rollback()`
 - 1 Não envia os dados para o SGBD, cancelando uma transação que estava em andamento.
 - 2 Uso: `con.rollback();`
- `void close()`
 - 1 Fecha um objeto de conexão com SGBD.
 - 2 Uso: `con.close();`

Métodos

- `java.sql.Statement.createStatement()`
 - 1 Usado para criação de uma instrução SQL.
 - 2 Uso: `Statement stmt = con.createStatement();`
- `java.sql.PreparedStatement.prepareStatement(String sql)`
 - 1 Para criação de uma instrução SQL parametrizada.
 - 2 Uso: `PreparedStatement pstmt = con.prepareStatement("select * from contas where id = ?");`

OBRIGADO!