

Algoritmos III

Swing com JDBC

Raphael de Souza, Karen Figueiredo

IC/UFMT

1 Swing com JDBC

- JDBC ou Java Database Connectivity
- Criando um conexão JDBC como BD
- Executando uma consulta com JDBC

JDBC ou Java Database Connectivity

Definição

É um conjunto de classes e interfaces (API) escritas em Java que fazem o envio de instruções SQL para qualquer banco de dados relacional

JDBC ou Java Database Connectivity

Definição

É um conjunto de classes e interfaces (API) escritas em Java que fazem o envio de instruções SQL para qualquer banco de dados relacional

- É necessário um driver específico para cada banco que deseja trabalhar
- Crie um projeto e adicione o driver baixado dentro da biblioteca do seu projeto

- É necessário um driver específico para cada banco que deseja trabalhar
- Crie um projeto e adicione o driver baixado dentro da biblioteca do seu projeto

- É necessário um driver específico para cada banco que deseja trabalhar
- Crie um projeto e adicione o driver baixado dentro da biblioteca do seu projeto

Criando um conexão JDBC como BD

```
Class.forName("org.postgresql.Driver");  
Connection con = DriverManager.getConnection("  
    jdbc:postgresql://localhost:5432/aula", "  
    postgres", "123");
```

```
jdbc:<SGBD>://<BANCO>:<PORTA>/<BANCO>  
jdbc:mysql://localhost:3306/aula  
jdbc:oracle:thin:@localhost:1521:aula  
jdbc:postgresql://localhost:5432/aula
```


Executando uma consulta com JDBC

- Pode-se realizar uma operação em um banco de dados com JDBC utilizando:
 - ▶ `java.sql.Statement`
 - ▶ `java.sql.PreparedStatement`
 - ▶ `java.sql.CallableStatement`

Executando uma consulta com JDBC

- Pode-se realizar uma operação em um banco de dados com JDBC utilizando:
 - ▶ `java.sql.Statement`
 - ▶ `java.sql.PreparedStatement`
 - ▶ `java.sql.CallableStatement`

Executando uma consulta com JDBC

- Pode-se realizar uma operação em um banco de dados com JDBC utilizando:
 - ▶ `java.sql.Statement`
 - ▶ `java.sql.PreparedStatement`
 - ▶ `java.sql.CallableStatement`

Executando uma consulta com JDBC

- Pode-se realizar uma operação em um banco de dados com JDBC utilizando:
 - ▶ `java.sql.Statement`
 - ▶ `java.sql.PreparedStatement`
 - ▶ `java.sql.CallableStatement`

Executando uma consulta com JDBC

- Pode-se realizar uma operação em um banco de dados com JDBC utilizando:
 - ▶ `java.sql.Statement`
 - ▶ `java.sql.PreparedStatement`
 - ▶ `java.sql.CallableStatement`

java.sql.Statement

Definição

Esta classe executa um código SQL a partir de uma String

Muito utilizado para consultas SQL estáticas

java.sql.Statement

Definição

Esta classe executa um código SQL a partir de uma String

Muito utilizado para consultas SQL estáticas

java.sql.Statement

Definição

Esta classe executa um código SQL a partir de uma String

Muito utilizado para consultas SQL estáticas

java.sql.Statement

```
Class.forName("org.postgresql.Driver");  
Connection conexao = DriverManager.getConnection(  
    "jdbc:postgresql://localhost:5432/aula", "  
    postgres", "123");  
  
Statement st = conexao.createStatement();  
st.executeUpdate("INSERT INTO pessoa (nome,cpf)  
    VALUES ('NOME', '000.000.000-00');");
```

java.sql.PreparedStatement

Definição

Esta classe executa um código SQL pré-compilado

java.sql.PreparedStatement

Definição

Esta classe executa um código SQL pré-compilado

java.sql.PreparedStatement

```
Class.forName("org.postgresql.Driver");
Connection conexao = DriverManager.getConnection(
    "jdbc:postgresql://localhost:5432/aula", "
    postgres", "123");

PreparedStatement pst = conexao.prepareStatement("
    INSERT INTO pessoa (nome,cpf) VALUES (?,?);");
pst.setString(1, "NOME");
pst.setString(2, "000.000.000-00");
pst.executeUpdate();
```

java.sql.PreparedStatement

Existem vários tipos de dados para setar no PreparedStatement

- setByte
- setBoolean
- setDate
- setDouble
- setFloat
- setInt
- setLong
- setShort
- etc...

java.sql.PreparedStatement

Existem vários tipos de dados para setar no PreparedStatement

- setByte
- setBoolean
- setDate
- setDouble
- setFloat
- setInt
- setLong
- setShort
- etc...

java.sql.PreparedStatement

- Pré-compila o código SQL
- Melhor desempenho
- Projete contra SQL Injection

java.sql.PreparedStatement

- Pré-compila o código SQL
 - Melhor desempenho
 - Projete contra SQL Injection

java.sql.PreparedStatement

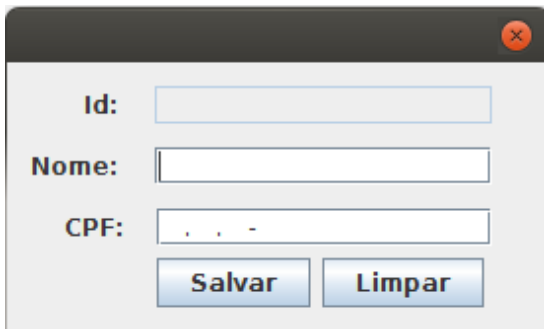
- Pré-compila o código SQL
- Melhor desempenho
- Projete contra SQL Injection

java.sql.PreparedStatement

- Pré-compila o código SQL
- Melhor desempenho
- Projete contra SQL Injection

Exercício

- Construa uma tela em Java, utilizando o pacote Swing, que corresponda a tela abaixo e implemente as ações dos botões da mesma utilizando o banco de dados PostgreSQL, utilizando tanto o Statement quanto o PreparedStatement



A screenshot of a Java Swing window with a dark title bar and a close button (red circle with an 'X'). The window contains three text input fields with labels to their left: 'Id:', 'Nome:', and 'CPF:'. The 'CPF:' field has pre-filled characters '. . -'. Below the input fields are two buttons: 'Salvar' and 'Limpar'.

Referências Bibliográficas

Aula baseada em slides preparados pelos seguintes professores:

- Prof. Bruno B. Boniati - www.cafw.ufsm.br/bruno - Universidade Federal de Santa Maria
- Prof. José Gustavo de Souza Paiva - <http://www.facom.ufu.br/jgustavo/disc/poo.html> - Faculdade de Computação da Universidade Federal de Uberlândia
- Prof. Vítor E. Silva Souza - <http://www.inf.ufes.br/vitorsouza/pt/> - Departamento de Informática da Universidade Federal do Espírito Santo

Livro:

- Guia de Estudo: Certificação Sun para Programador Java 6. Kathy Sierra e Bert Bates