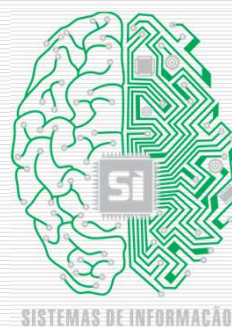


Prog. Orientada a Objetos II

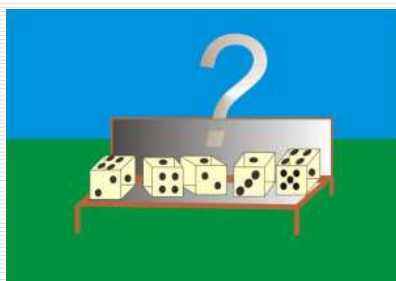
Prof. Fernando Del Moro
fernando@esucri.com.br



1

Prog. Orientada a Objetos II

Banco de Dados em Java



2

Prog. Orientada a Objetos II

Drivers

✓ Para desenvolver uma aplicação que acesse bancos de dados relacionais, é preciso ter disponível:

- ✓ Acesso ao servidor de banco de dados;
- ✓ Um *driver* adequado ao SGBD que será utilizado.

3

Prog. Orientada a Objetos II

Drivers

✓ Grande parte das linguagens utiliza um dos seguintes métodos para prover conexão:

- Acesso Nativo;
 - ✓ Utiliza bibliotecas específicas da linguagem
- Um *driver* ODBC (*Open Data Base Connectivity*).
 - ✓ Bibliotecas DLL fornecidas (normalmente pelo fabricante) para prover acesso

4

Prog. Orientada a Objetos II

Drivers no Java

- ✓ O Java não tem o recurso de Acesso Nativo, necessitando de *drivers*
- ✓ O Java pode fazer uso de dois tipos de *drivers*;
 - ODBC: mesmos *drivers* fornecidos a todos
 - JDBC: *drivers* escritos especificamente para Java

5

Prog. Orientada a Objetos II

O que é **JDBC** (**Java Database Connectivity**)?

- ✓ Para fazer a comunicação entre a aplicação e o SGBD é necessário possuir um *driver* para a conexão desejada.
- ✓ Geralmente, as empresas de SGBD oferecem o *driver* de conexão que seguem a especificação JDBC.

6

Prog. Orientada a Objetos II

O que é JDBC?

- ✓ É uma API que reúne conjuntos de classes e interfaces escritas em Java que possibilita a conexão a um banco através de um *driver* específico.
- ✓ Com esse *driver* pode-se executar instruções SQL em qualquer tipo de banco de dados relacional.

7

Prog. Orientada a Objetos II

Vantagens

- ✓ A programação do sistema é a mesma para qualquer SGBD, as aplicações não são voltadas (e amarradas) a um BD específico
- ✓ Mantém a independência de plataforma da linguagem Java
- ✓ Aplicações rodam em “qualquer” SO acessando “qualquer” BD

8

Prog. Orientada a Objetos II

Características

- ✓ A API traz encapsulado:
 - ✓ O estabelecimento da conexão com o BD
 - ✓ O envio de comandos SQL
 - ✓ O processamento dos resultados

9

Prog. Orientada a Objetos II

JDBC x ODBC

- ✓ ODBC tem o mesmo propósito que o JDBC e existe para várias plataformas então por que não usar ODBC?
- ✓ Os *drivers* ODBC são escritos em C, o que limita a portabilidade dos programas Java
- ✓ O ODBC tem que ser instalado e configurado na máquina do cliente

10

Prog. Orientada a Objetos II

JDBC x ODBC

- ✓ Por serem *drivers* genéricos, podem trazer implementações desnecessárias.
- ✓ Apesar disso, o Java pode utilizar os recursos ODBC para acessar bancos de dados, mas é algo desaconselhado.

11

Prog. Orientada a Objetos II

SGBD	Formato da URL de conexão
	<code>jdbc:oracle:thin:@host:porta:nomeBD</code> (Porta padrão 1521)
	<code>jdbc:postgresql://host:porta/nomeBD</code> (Porta padrão 5432)
	<code>jdbc:db2://host:porta/nomeBD</code> (Porta padrão 446, 6789, ou 50000)
	<code>jdbc:sqlserver://host:porta;databaseName=nomeBD</code> (Porta padrão 1433)
	<code>jdbc:mysql://host:porta/nomeBD</code> (Porta padrão 3306)
	<code>jdbc:firebirdsql:host/porta/caminhoArquivo.fdb</code> (Porta padrão 3050)
	<code>jdbc:sqlite:caminhoArquivo</code>
	<code>jdbc:odbc:nomeBD</code>

12

Prog. Orientada a Objetos II

✓ O *NetBeans* traz na sua instalação padrão *Drivers JDBC* para: *Derby Java DB* e *PostgreSQL*.

