Pergunta 1

0,25 em 0,25 pontos

O código a seguir deve acessar a Tabela "aluno" de um Banco de Dados e mostrar no console do Java o nome de todos os alunos nela cadastrados. Complete as lacunas do código a seguir escolhendo uma das alternativas apresentadas com sua respectiva ordem de acordo com as lacunas numeradas no código fonte.

```
String query = "select * from aluno";
try {
 Class.forName(driver);
 _____(1)_____ x = DriverManager._____(2)_____ (server, user, senha);
 _____(3)_____ y = x.prepareStatement(query);
 _____(4)_____ z = y.executeQuery();
 while (z.next()) {
  String nome = z.getString("nome");
  System.out.println("Aluno: " + nome);
 }
 y.close();
 x.close();
} catch (Exception e) {
 e.printStackTrace();
}
```

Resposta Selecionada: e. Connection, getConnection, Statement, ResultSet

Respostas:

a. Statement, getStatement, Connection, executeQuery

b. Statement, getConnection, Statement, Query

c. Connection, getServer, Statement, ResultSet

d. Connection, getStatement, Statement, Query

e. Connection, getConnection, Statement, ResultSet

Comentário da resposta: Resposta: E
Comentário: no "espaço 1" do programa, algum elemento (alguma classe cujo objeto será representado pelo "x") está recebendo algo que é recuperado a partir de um Driver Manager. Vimos que a classe DriverManager é responsável por solicitar uma conexão com o Banco de Dados. Desta forma, o retorno de um método dessa classe deve ser a conexão, ou seja, uma classe do tipo Connection. Vimos também que o método que permite como retorno uma conexão com o Banco de Dados é o método getConnection(...), em que colocamos como parâmetro os dados de localização, usuário e senha do Banco de Dados. O "espaço 3" deve receber uma classe, representada pelo objeto "y", que recebe o retorno de um método "prepareStatement(...)", ou seja, um objeto do tipo Statement. O "espaço 4" é um elemento de retorno de um método "executeQuery()". Vimos que o retorno desse método é o conteúdo lido do Banco de Dados, de forma que será recebido por um objeto do tipo ResultSet.

Pergunta 2

0,25 em 0,25 pontos

Qual o método da classe Statement (da biblioteca java.sql da API do Java DataBase Connectivity – JDBC) que possibilita acessar e ler registros do Banco de Dados?

Resposta Selecionada: c. executeQuery ( )

Respostas:

a. executeCommand ( )

b. execute ( )

c. executeQuery ( )

d. executeReg ( )

e. getRegister ( )

Comentário da resposta: Resposta: C
Comentário: vimos que é possível realizar dois tipos de acesso a dados do Banco de Dados, um acesso para alteração de informação e outro para a leitura das informações. O primeiro tipo será efetuado utilizando-se o método "executeUpdate(...)" da classe de Statement. Já o segundo tipo é efetuado utilizando-se o método "executeQuery(...)" da classe de Statement. Desta forma, o enunciado pede o acesso para a leitura dos registros, o que exige a utilização do método "executeQuery(...)".

Pergunta 3

0,25 em 0,25 pontos

Os *softwares* de Bancos de Dados como, por exemplo, Oracle, SQLServer e MySQL, utilizados para persistência de dados, são frequentemente chamados de Sistemas Gerenciadores de Banco de Dados (SGDB). Tais sistemas são encarregados de realizar as operações que o usuário necessitar realizar nas diversas tabelas do banco. No que se refere ao JDBC, avalie as seguintes afirmativas:

I - O JDBC permite acesso a bancos de dados relacionais.
II - O JDBC permite que os programas invoquem *procedures* armazenados, a partir de objetos que implementam a interface ResultSet.
III - O programador pode carregar um *drive* JDBC específico para um Banco de Dados utilizando a classe java.sql.DriverManager, que busca estabelecer uma ligação com o Banco de Dados pelo método getConnection().
IV - No JDBC, os problemas de acesso ao Banco de Dados são tratados como exceções.
V - Um programa deve primeiro se conectar ao Banco de Dados para então carregar o *driver* desse Banco de Dados.

Assinale a alternativa correta:

Resposta Selecionada: a. I, III e IV estão corretas.

Respostas:
a. I, III e IV estão corretas.

b. I e II estão corretas.

c. II e III estão corretas.

d. II, IV e V estão corretas.

e. I e V estão corretas.

Comentário da resposta: Resposta: A
Comentário: a afirmação I está correta, já que mostra o objetivo principal da API JDBC, que é o acesso a SGBDs relacionais. A afirmação II está incorreta, já que contém uma explicação incorreta sobre o "ResultSet", que é uma classe utilizada para receber toda informação lida e retornada do Banco de Dados (não podendo ser utilizada, por exemplo, quando vamos alterar informações do BD). A afirmação III está correta quando explica a função da classe DriverManager, cuja finalidade é a de fornecer uma conexão direta com o Banco de Dados, a partir do método getConnnection(...) da classe DriverManager. A afirmação IV explica que as classes de acesso ao Banco de Dados se utilizam das exceções descendentes da classe "SQLException" que por sua vez descende da classe Exception. Esse fato explica a obrigatoriedade do tratamento de exceções (com a estrutura "try catch"), toda vez que se vai utilizar os procedimentos de acesso a BDs, tratando os possíveis problemas resultantes dessa tentativa de acesso, terminando por definir e caracterizar esses problemas como exceções. A afirmação V está incorreta e acaba por propor o processo de forma contrária, já que precisamos primeiro carregar (localizar) o *driver* do Banco de Dados e então utilizá-lo para gerar a conexão de acesso com o BD.

Pergunta 4

0,25 em 0,25 pontos

Sobre acesso a Banco de Dados, analise as seguintes afirmativas:
I – Para que possamos acessar um Banco de Dados específico, devemos ter o arquivo de *driver* desse Banco de Dados e deve ser importado no projeto e apontado pelo DriverManager para gerar a conexão.
II – No momento da conexão, ela é feita diretamente a uma Base existente no Banco de Dados, a partir de um Usuário e de uma Senha fornecidos pelo DBA.
III – O Statement é a classe capaz de guardar os dados do Banco de Dados, já que é a partir dela que executamos as *queries* que buscam esses dados.
IV – Os dados guardados no Statement podem ser acessados como se fossem ponteiros, em que apontamos para cada linha de dado obtida do BD.

Assinale a alternativa correta:

Resposta Selecionada: d. I e II estão corretas.

Respostas:
a. II, III e IV estão corretas.

b. I, II e III estão corretas.

c. IV está correta.

d. I e II estão corretas.

e. Todas as afirmações estão corretas.

Comentário da resposta: Resposta: D
Comentário: a afirmação I está correta e explica o objetivo principal da utilização de um *driver* JDBC, que é possibilitar o acesso a determinados BDs. Desta forma, para que o programa Java tenha acesso a esse *driver*, este deve ser importado ao projeto a partir das "Propriedades do Projeto" (no seu desenvolvimento). A afirmação II está correta quando afirma que a conexão é feita diretamente a uma Base de Dados existente no banco. Lembre-se de que uma das informações que colocamos na URL de acesso a BDs é o nome da Base de Dados, além de sabermos que na geração da conexão colocamos na String: o nome e a senha do Usuário que possui acesso àquela Base de Dados definida na URL de acesso. Tanto a afirmação III quanto a afirmação IV explicam na verdade o conceito da classe ResultSet, que é a classe responsável por receber as informações lidas do BD, guardando-as de uma maneira cujo acesso a elas é realizado a partir de manejamento de ponteiros (como com a utilização do método "next(...)").

Pergunta 5

0,25 em 0,25 pontos

Seja "Janela" uma classe que herda a classe JFrame da biblioteca Swing. A instrução "this.addWindowListener(this);" indica que:

Resposta Selecionada: b.
A própria classe foi adicionada à sua "lista de ouvintes de eventos de janela" (do tipo WindowEvent), da qual será a controladora dos eventos.

Respostas:

a. Foi feita uma adição de uma "lista de ouvintes" aos diversos componentes do *Frame*.

b.
A própria classe foi adicionada à sua "lista de ouvintes de eventos de janela" (do tipo WindowEvent), da qual será a controladora dos eventos.

c. Foi feito um registro de um ouvinte para os eventos de ação dos componentes da janela.

d. Trata-se de um controlador de ouvintes para eventos de ação dos botões da Janela.

e. Foi criado um método na Classe a fim de registrar os seus possíveis ouvintes.

Comentário da resposta: Resposta: B
Comentário: a instrução "this.addWindowListener(this);" é utilizada quando precisamos controlar alguns eventos de janela (como o fechamento, a minimização, entre outros). Esse método da classe JFrame é uma instrução que prepara um elemento para ser o controlador de todos os eventos de janela que podem ocorrer. Esse controlador será o responsável por "escutar" o evento e dar andamento ao procedimento programado (codificado) quando na ocorrência daquele evento. Além disso, o método aqui comentado define em seu parâmetro qual elemento será o responsável pelo gerenciamento do evento, que no caso possui a palavra "this", significando que a classe do JFrame irá se autogerenciar.

Pergunta 6

0,25 em 0,25 pontos

Dos métodos listados nos itens a seguir, qual deles é utilizado para adicionar um "observador" que verifica se um determinado botão (um componente interno ao *container*) foi apertado ("clicado"), a partir da biblioteca do AWT ou Swing, passando como parâmetro o operador de referência da classe?

Resposta Selecionada: a. botao.addActionListener(this)

Respostas:

a. botao.addActionListener(this)

b. botao.addListener(this)

c. botao.addMouseListener(this)

d. botao.addWindowListener(this)

e. botao.addEventListener(this)

Comentário da resposta: Resposta: A
Comentário: vimos que os eventos acionados pelo usuário são controlados no Java por interfaces do tipo Listener. Sendo assim, para controlar o evento de um usuário que aperta (aciona) um botão existente na tela, ou num formulário, o Listener responsável é o "ActionListener", de forma que para que o botão funcione efetivamente, ou seja, para que o programa perceba que ele foi acionado pelo usuário, esse Listener deve ser adicionado ao botão, pois do contrário ele não será percebido e consequentemente não gerará nenhuma ação após seu acionamento.

Pergunta 7

0,25 em 0,25 pontos

Em JDBC, o que é correto afirmar sobre a classe "Statement"?

Resposta Selecionada: e. É um canal de comunicação utilizado para submeter uma ação (query) em um Banco de Dados (consulta ou alteração).

Respostas:

a. É utilizada para obter uma conexão (Connection) com um banco de dados, por meio da URL, do *LOGIN* e da SENHA.

b. É utilizada para incluir um *driver* de conexão com um Banco de Dados em uma aplicação em Java.

c. É um ResultSet utilizado para armazenar os dados retornados de um Banco de Dados.

d. Possibilita armazenar comandos SQL pré-compilados ou pré-processados no Banco de Dados.

e. É um canal de comunicação utilizado para submeter uma ação (query) em um Banco de Dados (consulta ou alteração).

Comentário da resposta: Resposta: E
Comentário: vimos que a classe Statement é a classe que possui os métodos executeUpdate(...) e executeQuery(...) que permitem a execução de ações diretas no BD (com as *queries* definidas em seus parâmetros). Desta forma, o papel da classe Statement no programa Java é o de estabelecer o canal de comunicação (de duas vias) com o BD, permitindo o envio de uma *query*, além do recebimento das informações eventualmente lidas no Banco de Dados e, nesse caso, colocando essas informações em um objeto do tipo ResultSet.

Pergunta 8

0,25 em 0,25 pontos

Sobre acesso a Banco de Dados, analise o código a seguir:
Complete as lacunas do código a seguir escolhendo uma das alternativas apresentadas com sua respectiva ordem de chamada dos métodos no código fonte.

```
...
try {
 Class.forName("com.mysql.jdbc.Driver");
 con = DriverManager._____("jdbc:mysql://localhost:3306/aula_alpoo", "root", "root");
 PreparedStatement st = con._____("SELECT * FROM aluno");
 ResultSet rs = st._____();
 while (rs._____()) {
  String nome = rs._____("nome");
  System.out.println("Aluno: " + nome);
 }
 con.close();
} catch (Exception e) {
 e.printStackTrace();
}
...
```

Resposta Selecionada: c. getConnection, prepareStatement, executeQuery, next, getString

Respostas:

a. prepareStatement, getConnection, executeQuery, getNext, parse

b. getConnection, getStatement, execute, next, getValue

c. getConnection, prepareStatement, executeQuery, next, getString

d. prepareStatement, connect, executeQuery, getNext, parseString

e. setDrive, connect, execute, next, setString

Comentário da resposta: Resposta: C
Comentário: analisando as lacunas representadas pelos espaços sublinhados do programa do enunciado, temos que a primeira lacuna representa algo que pertence à classe DriveManager, como o método "getConnection(...)", já que as informações do parâmetro mostrado na linha de comando do enunciado são dados como a localização e o usuário com permissão de acesso às bases de dados do BD. A segunda lacuna requer algum elemento que indica um método de alguma classe, já que é seguido de um parâmetro. Como o retorno desse método deve ser um objeto do tipo Statement, então o método deve ser o prepareStatement(...). Percebe-se que o valor do parâmetro é uma String de uma "query" de acesso a dados em um BD (um "select"). Como a *query* é um *select*, a terceira lacuna é um método que retorna um objeto do tipo ResultSet, ou seja, é o método executeQuery. A quarta e quinta lacunas se referem à leitura do ResultSet, o qual para que seja lido valor por valor, deve-se utilizar o processo de leitura "linha a linha" (ou "dado a dado"), movendo-se o ponteiro sempre ao próximo valor, com o método "next". A última lacuna é uma forma de se capturar a informação do ResultSet já no formato da variável que vai receber a informação, que no caso é uma String, e por isso utilizou-se o método "getString(...)".

Pergunta 9

0,25 em 0,25 pontos

Podemos afirmar que o método "execute" da Classe PreparedStatement:

Resposta Selecionada: b.
Executa uma Query SQL no Banco de Dados sem retornar nenhum dado específico (como quando utilizado para o comando sql: "DELETE").

Respostas:

a.
Faz com que seja executada uma Query SQL no Banco de Dados, por meio de uma conexão já estabelecida, retornando um conjunto de dados em uma classe do tipo ResultSet.

b.
Executa uma Query SQL no Banco de Dados sem retornar nenhum dado específico (como quando utilizado para o comando sql: "DELETE").

c.
É um método que deve ser implementado na Classe criada para acesso a Banco de Dados, já que ela implementa a Interface "sqlConnector".

d. Somente pode ser utilizado em Classes que herdam a Classe Frame.

e.
Estabelece uma conexão com o Banco de Dados, para que seja possível utilizá-lo em processos de captura de dados de um BD.

Comentário da resposta: Resposta: B
Comentário: o método "execute(...)" da classe PreparedStatement equivale ao método "executeUpdate(...)" da classe Statement, ou seja, é um método utilizado pelo canal de comunicação (no caso pelo objeto do tipo PreparedStatement) para realizar alterações nas informações de uma tabela do Banco de Dados, de forma que esse método não é utilizado para leitura de informação, já que ele não retorna informações para um objeto do tipo ResultSet, lembrando que as *queries* que realizam alterações no BD são as *queries* "delete", "insert" e "update".

Pergunta 10

0,25 em 0,25 pontos

A partir de um programa (*desktop*) a ser criado na linguagem Java, no qual deve-se gerar um Formulário possível de ser preenchido pelo Usuário, o que podemos afirmar sobre "eventos" e como eles devem ser tratados pelo desenvolvedor do sistema?

Resposta Selecionada: d.
São ações realizadas pelo usuário (no sistema), o qual espera uma resposta do sistema. No Java, eles são tratados a partir das interfaces "Listeners".

Respostas:

a. São os objetivos do usuário na empresa que trabalha e devem ser tratados com muita cautela.

b.
São as posições dos componentes da tela e são tratados como "retângulos individuais", em que o ponto de referência é o ponto superior direito do componente.

c.
É a ordem de abertura das janelas dos formulários dependendo do que o usuário pretende fazer, de forma que essa ordem depende da sequência com que é dada (no código) a visibilidade dos *Frames*.

d.
São ações realizadas pelo usuário (no sistema), o qual espera uma resposta do sistema. No Java, eles são tratados a partir das interfaces "Listeners".

e. São os acessos ao Banco de Dados, o qual é programado a partir das APIs de JDBC.

Comentário da resposta: Resposta: D
Comentário: quando um usuário está utilizando um programa criado para *desktop* na linguagem Java, ele interage com os componentes da tela (campos, botões etc.), esperando a cada ação realizada uma reação do sistema (como a ação de fechar o sistema se o usuário seleciona a opção "Sair" do menu do sistema). Para que o sistema reaja a cada ação do usuário, esse controle do sistema foi desenvolvido utilizando-se os elementos de "Listeners" disponíveis na biblioteca da linguagem, que são as "interfaces" que permitem esse controle.

← OK