



Universidade Eduardo Mondlane

Faculdade de Engenharia

Departamento de Engenharia Electrotécnica

Curso de Engenharia Informática

**Guião de Correção: Teste 2 de Programação Orientada a Objectos II (11/11/2015)**

**Duração: 150 minutos**

1. Suponha que é dado o formulário da figura a seguir:

Código da Agência:	<input type="text"/>	Número da Conta:	<input type="text"/>
Nome:	<input type="text"/>		
Endereço:	<input type="text"/>		
Telefone:	<input type="text"/>		
CPF:	<input type="text"/>		
<input checked="" type="radio"/> Conta Corrente <input type="radio"/> Conta Poupança			
<input type="button" value="Consultar"/> <input type="button" value="Atualizar"/> <input type="button" value="Fechar"/>			

Supondo que é dado uma classe Modelo “DadosPessoais” que contem toda informação do formulário acessíveis através de getters e setters e que tenha uma classe BDconexao que tenha um método getConnection() que devolve uma conexão a Base de dados:

- Crie um método que permite **inserir** os dados do formulário numa tabela de base de dados chamada “DadosPessoais”; **(2V)**
- Crie um Método que permita encontrar 1 cliente pelo Numero de Conta; **(1.5V)**
- Crie um Método que permita encontrar uma lista de clientes pelo Código da agencia; **(2V)**
- Crie um Método que permita actualizar as informações de uma conta. **(2V)**

a) Metodo inserir

```
public void inserir(DadosPessoais dp) throws SQLException{
    String sql = "INSERT INTO dadosPessoais(codAgencia,nome,endereco,"
        + "telefone,cpf,tipo) VALUES(?,?,?,?,?,?,?)";
    PreparedStatement st = conexao.prepareStatement(sql);
    st.setString(1, dp.getCodAgencia());
    st.setString(2, dp.getNome());
    st.setString(3, dp.getEndereco());
    st.setString(4, dp.getTelefone());
    st.setString(5, dp.getCpf());
    st.setString(6, dp.getTipoDeConta());
    st.executeUpdate();
}
```

b) Encontrar Cliente por numero de conta

```
public DadosPessoais encontrarPorId (int id) throws SQLException{
    String sql = "SELECT * FROM dadosPessoais WHERE id = ?";
    PreparedStatement st = conexao.prepareStatement(sql);
    st.setInt(1, id);
    ResultSet rs = st.executeQuery();
    if(rs.next()){
        DadosPessoais dp = new DadosPessoais();
        dp.setCodAgencia(rs.getString("codAgencia"));
        dp.setCpf(rs.getString("cpf"));
        dp.setEndereco(rs.getString("endereco"));
        dp.setNome(rs.getString("nome"));
        dp.setNumeroDaConta(rs.getInt("id"));
        dp.setTelefone(rs.getString("telefone"));
        dp.setTipoDeConta(rs.getString("tipo"));
        return dp;
    }
    return null;
}
```

**Bom trabalho**  
**Eng. Ruben Manhiça**

c) Lista de clientes

```
public List<DadosPessoais> encontrarPorAgencia (String codAgencia) throws SQLException{
    String sql = "SELECT * FROM dadosPessoais WHERE codAgencia = ?";
    PreparedStatement st = conexao.prepareStatement(sql);
    st.setString(1, codAgencia);
    ResultSet rs = st.executeQuery();
    List<DadosPessoais> lista = new ArrayList<>();
    while(rs.next()){
        DadosPessoais dp = new DadosPessoais();
        dp.setCodAgencia(rs.getString("codAgencia"));
        dp.setCpf(rs.getString("cpf"));
        dp.setEndereco(rs.getString("endereco"));
        dp.setNome(rs.getString("nome"));
        dp.setNumeroDaConta(rs.getInt("id"));
        dp.setTelefone(rs.getString("telefone"));
        dp.setTipoDeConta(rs.getString("tipo"));
        lista.add(dp);
    }
    return lista;
}
```

d) Actualizar

```
public void actualizar(DadosPessoais dp) throws SQLException{
    String sql = "UPDATE dadosPessoais SET codAgencia=?,nome=?,endereco=?, "
        + "telefone=?,cpf=?,tipo=? WHERE id = ?";
    PreparedStatement st = conexao.prepareStatement(sql);
    st.setString(1, dp.getCodAgencia());
    st.setString(2, dp.getNome());
    st.setString(3, dp.getEndereco());
    st.setString(4, dp.getTelefone());
    st.setString(5, dp.getCpf());
    st.setString(6, dp.getTipoDeConta());
    st.setInt(7, dp.getNumeroDaConta());
    st.executeUpdate();
}
```

2. É Dado o seguinte modelo de relatório em Jasper: (7.5V)

CLIENTES EM CREDITO					
new java.util.Date()					
Numero	Nome	Responsavel	Contactos	Endereço	Em Credito
\$F	\$F{nome}	\$F{responsavel}	\$F{contacto}	\$F{endereco}	\$F{emCredito}
Page Footer					
"Page "+\$V" " + \$V					

Crie uma classe em Java que carregue dados para este relatório. Nesta classe deverá ter uma classe **ClientesJBean** que irá servir de datasource para os campos da tabela apresentada e gere a seguinte informação para o relatório

1	Milagre	Joana	820000012	Alto Mae	0.0
2	Antonio	Catia	870000011	Baixa	1000
3	Daniel	Ana	860000011	Malhazine	2000
4	Rafael	Leticia	840000011	Zimpeto	300
5	Pedro	Patricia	820000011	Matendene	500

```

public class GerarRelatorio {

    public void gerarRelatorio() throws JRException {
        List<ClienteJbean> clientes = new ArrayList<>();
        ClienteJbean cj1 = new ClienteJbean(1, "Milagre", "Joana", "820000012", 0.0);
        clientes.add(cj1);
        ClienteJbean cj2 = new ClienteJbean(2, "Antonio", "Catia", "870000011", 1000);
        clientes.add(cj2);
        ClienteJbean cj3 = new ClienteJbean(3, "Daniel", "Ana", "860000011", 2000);
        clientes.add(cj3);
        ClienteJbean cj4 = new ClienteJbean(4, "Rafael", "Leticia", "840000011", 300);
        clientes.add(cj4);
        ClienteJbean cj5 = new ClienteJbean(5, "Pedro", "Patricia", "820000011", 500);
        clientes.add(cj5);
        Map parametros = new HashMap();
        JRBeanCollectionDataSource dataSource = new JRBeanCollectionDataSource(clientes);
        InputStream ficheiro = getClass().getResourceAsStream("/Exercicio2/ClientesEmCredito.jasper");
        JasperPrint jasperPrint = JasperFillManager.fillReport(ficheiro, parametros, dataSource);
        JasperViewer jasperViewer = new JasperViewer(jasperPrint, true);
        jasperViewer.setVisible(true);
    }
}

```

3. Sobre o seu projecto responda as questões:

a) Qual foi a versão do JDK usado? (1 Valor)

**1.8; 1.7; 1.6; etc...**

b) No seu projecto semestral por diversas ocasiões teve de fazer tratamento a diversos tipos de erros funcionais, recorrendo ao try...catch ou fazendo throws Exception, indique as diferenças entre ambas formas de tratamento de erros, e em que parte do projecto optou por um e não por outra. (1 Valor)

**Try catch é usado para tratar uma exceção no local do código em que ela ocorre; throws Exception lança a responsabilidade de tratar a exceção a classe que invocar o método.**

c) Explique a diferença de Swing e JDBC e diga em que parte do seu projecto cada um deles foi importante e indicando a respectiva camada em que ele pode ser usado. (1 Valores)

**Swing é uma biblioteca java para desenho de interfaces e usamos na camada de apresentação; JDBC é um driver para comunicação com Base de dados e usamos na camada de acesso a dados**

d) No seu projecto por diversas vezes teve de passar dados de um formulário para o outro, explique em poucas palavras como conseguiu-o fazer. (2 Valores)

**Varias Possibilidades**

**Acceptáveis:**

- **Passagem de valores por parâmetro do construtor;**
- **Definição de variáveis como estáticas;**