



**Universidade Eduardo Mondlane**

**Faculdade de Engenharia**

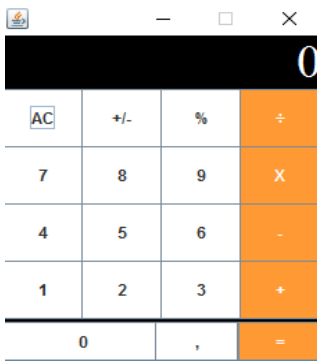
**Departamento de Engenharia Electrotécnica**

**Curso de Engenharia Informática**

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

**Duração: 150 minutos**

1. Suponha que é dado a calculadora da figura a seguir:



Todos os métodos desta calculadora já foram implementados, e ela realiza todas operações dadas. Agora supondo que tenha uma tabela de base de dados chamada **Historico** (numero1, numero2, operação, resultado, data) implementada em java através de uma classe de nome **Historico**. Suponha que já tenha também uma classe **BDconexao** e que nela já tenha um método **getConnection()** que devolve uma conexão ao banco de dados.

Resolva os seguintes problemas:

a) Crie uma classe modelo de acesso aos dados armazenados na tabela histórico, realizando 1 operação de inserção, leitura e actualização de dados; (2V)

```
public void inserir(Historico dp) throws SQLException{
    String sql = "INSERT INTO historico (numero1,numero2,"
        + "operacao,resultado,data) VALUES(?, ?, ?, ?, ?) ";
    PreparedStatement st = conexao.prepareStatement(sql);
    st.setDouble(1, dp.getNumero1());
    st.setDouble(2, dp.getNumero2());
    st.setString(3, dp.getOperacao());
    st.setDouble(4, dp.getResultado());
    st.setDate(5, new Date(dp.getData().getTime()));
    st.executeUpdate();
}

public Historico encontrarPorId (int id) throws SQLException{
    String sql = "SELECT * FROM historico WHERE id = ?";
    PreparedStatement st = conexao.prepareStatement(sql);
    st.setInt(1, id);
    ResultSet rs = st.executeQuery();
    if(rs.next()){
        Historico dp = new Historico();
        dp.setData(new java.util.Date(rs.getDate("data").getTime()));
        dp.setId(rs.getInt("id"));
        dp.setNumero1(rs.getDouble("numero1"));
        dp.setNumero2(rs.getDouble("numero2"));
        dp.setOperacao(rs.getString("operacao"));
        dp.setResultado(rs.getDouble("resultado"));
        return dp;
    }
    return null;
}
```

```

public List<Historico> listarTodos () throws SQLException{
    String sql = "SELECT * FROM historico";
    PreparedStatement st = conexao.prepareStatement(sql);
    ResultSet rs = st.executeQuery();
    List<Historico> lista = new ArrayList<>();
    while(rs.next()){
        Historico dp = new Historico();
        dp.setData(new java.util.Date(rs.getDate("data").getTime()));
        dp.setId(rs.getInt("id"));
        dp.setNumero1(rs.getDouble("numero1"));
        dp.setNumero2(rs.getDouble("numero2"));
        dp.setOperacao(rs.getString("operacao"));
        dp.setResultado(rs.getDouble("resultado"));
        lista.add(dp);
    }
    return lista;
}

public void atualizar(Historico dp) throws SQLException{
    String sql = "UPDATE historico SET numero1=?,numero2=?, "
        + "operacao=?,resultado=?,data=? WHERE id=?";
    PreparedStatement st = conexao.prepareStatement(sql);
    st.setDouble(1, dp.getNumero1());
    st.setDouble(2, dp.getNumero2());
    st.setString(3, dp.getOperacao());
    st.setDouble(4, dp.getResultado());
    st.setInt(6, dp.getId());
    st.setDate(5, new Date(dp.getData().getTime()));
    st.executeUpdate();
}

```

- b) Se for a realizar uma consulta nesse banco de dados do tipo listar, no Resultset que teria como retorno, como é que referenciaria o campo “data” da base de dados? (1.5V)

**Supondo que tenha um objecto de nome, resultset teria:**

**resultset.getDate(“data”)**

- c) Crie um método listarOperacoes(Date inicio, Date fim), que apresenta todas operações que foram realizadas no intervalo de datas. (2V)

```

public List<Historico> listarOperacoes(java.util.Date inicio, java.util.Date fim) throws SQLException{
    String sql = "SELECT * FROM historico WHERE data BETWEEN ? AND ?";
    PreparedStatement st = conexao.prepareStatement(sql);
    st.setDate(1, new Date(inicio.getTime()));
    st.setDate(2, new Date(fim.getTime()));
    ResultSet rs = st.executeQuery();
    List<Historico> lista = new ArrayList<>();
    while(rs.next()){
        Historico dp = new Historico();
        dp.setData(new java.util.Date(rs.getDate("data").getTime()));
        dp.setId(rs.getInt("id"));
        dp.setNumero1(rs.getDouble("numero1"));
        dp.setNumero2(rs.getDouble("numero2"));
        dp.setOperacao(rs.getString("operacao"));
        dp.setResultado(rs.getDouble("resultado"));
        lista.add(dp);
    }
    return lista;
}

```

- d) Crie um método guardar operação que estará no JFrame principal do programa e irá armazenar a operação no Banco de dados; (2V)

```
public void guardar (String operacao, double numero1, double numero2,
    double resultado, java.util.Date data) throws SQLException, ClassNotFoundException{
    Historico h = new Historico(operacao, numero1, resultado, data);
    Exercicio1DAO edao = new Exercicio1DAO();
    edao.inserir(h);
}
```

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

RELATORIO DE OPERAÇÕES DO FUNCIONARIO						
Nome:	\$P{nome}			De	\$P{dataInicio}	á \$P{dataFim}
Contacto:	\$P{contacto}			new java.util.Date()		
Endereco:	\$P{endereco}					
Table Header						
IdOperacao	Tipo	CodMpesa	Sistema	Data	Cliente	Valor
\$F{idOperacao}	\$F{tipo}	\$F{codMpesa}	\$F{sistema}	\$F{data}	\$F{cliente}	\$F{valor}
Page Footer						"Page "+\$V" " + \$V

Crie uma classe em Java que carregue dados para este relatório. Nesta classe deverá ter uma classe **FuncionarioJBean** que irá servir de datasource para os campos da tabela apresentada e gere automaticamente 5 linhas para essa tabela.

```
public class GerarRelatorio {

    public void gerarRelatorio() throws JRException {

        List<FuncionarioJBean> beans = new ArrayList<>();
        for(int i = 1; i<=5; i++){
            beans.add(new FuncionarioJBean(i, "tipo: "+i, "MP"+i, "MeuSistema", "Cliente: "+i, Math.random()*1000));
        }
        Map parametros = new HashMap();
        parametros.put("nome", "Pedro Antonio");
        parametros.put("contacto", "845569321");
        parametros.put("endereco", "Avenida de Moçambique");
        parametros.put("dataInicio", "17/11/2015");
        parametros.put("dataFim", "30/11/2015");
        JRBeanCollectionDataSource dataSource = new JRBeanCollectionDataSource(beans);
        InputStream ficheiro = getClass().getResourceAsStream("/Exercicio2/OperacoesFuncionario.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) Qual/Quais é/são a/as biblioteca(s) usada(s) para o desenho das interfaces de utilizador em Java no seu projecto semestral; (1 Valor)

**Swing, AWT, SwingX, JCalendar, etc**

- c) Qual é o driver que usou para conexão a base de dados e qual é a principal excepção que pode ser lançada pelo uso dele. (1 Valor)

**Driver: JDBC**

**Excepção: SQLException**

- d) Quais são as ferramentas aprendidas na cadeira para geração de relatórios? Qual ferramenta usou e quais foram os principais factores que influenciaram nessa escolha? (1 Valor)

**Itext e ireports**

- e) Diferencie Método de Evento em Java e de exemplos de alguns Eventos que teve de ter no seu projecto semestral. (1 Valor)

**Bom trabalho**

**Eng. Ruben Manhiça**