ROTEIRO AULA PRÁTICA

Unidade 1 | Seção 3

Programação orientada ao objeto II

ROTEIRO DE AULA PRÁTICA

NOME DA DISCIPLINA: Programação orientada ao objeto II

Unidade 1

Seção 1.3

OBJETIVOS
Definição dos objetivos da aula prática:
Criar um CRUD usando MySql e Java. A atividade foi elaborada em duas etapas, na primeira é criada a interface do sistema e a segunda o banco de dados e as respectivas funcionalidades.

INFRAESTRUTURA	
Instalações:	
Laboratório de computação	
Materiais de consumo:	
Descrição	Quantidade de materiais por procedimento/atividade
Computadores	Máximo dois alunos por computador
Software:	
Sim (X) Não ( ) Em caso afirmativo, qual? _ Eclipse OXYGEN .2 e Java SDK 9____. Pago ( ) Não Pago (X) Tipo de Licença: ___ Eclipse Public License e Oracle Binary Code License _____.	
Descrição do software:	
O Eclipse OXYGEN .2 deve ser instalado com a opção: “Eclipse IDE for Java Developers” O java SDK deve ser instalado de forma padrão	
Equipamento de Proteção Individual (EPI):	
Não se aplica.	

PROCEDIMENTOS PRÁTICOS

Neste momento você deve ajudar o professor por meio da descrição de todas as etapas que deverão ser realizadas para a execução dos procedimentos práticos. Considerando a carga horária da aula prática, você pode replicar a caixa de procedimento/atividade quantas vezes for necessário.

Procedimento/Atividade n.1

Atividade proposta:

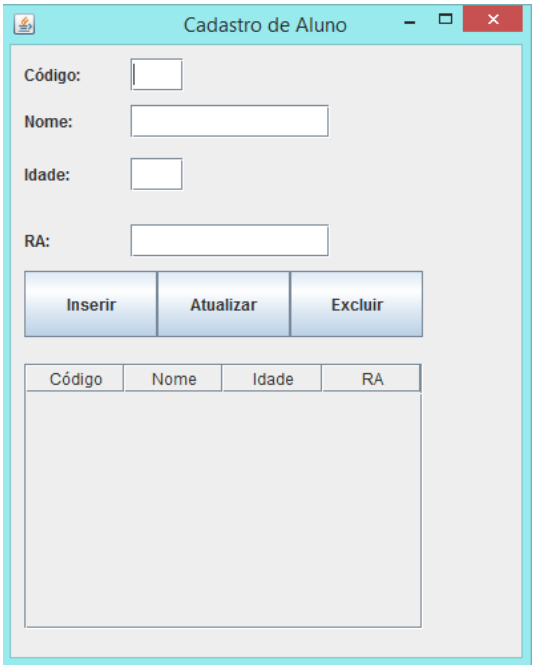
Criando uma *interface* gráfica e um CRUD para cadastro de aluno

Procedimentos para a realização da atividade:

Nessa primeira etapa o objetivo é criar a interface gráfica para cadastrar/excluir/alterar um aluno. Para isso, crie um novo projeto Java chamado “CadastroAluno”.

Crie uma nova classe chamada “InterfaceGrafica” e implemente o código necessário para gerar a interface gráfica da Figura 1:

Figura 1 – Resultado da interface gráfica



Fonte: Captura de tela da IDE Eclipse.

Caso tenha dúvidas, consulte o código baixo para implementar.

```
import java.awt.Container;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.sql.ResultSet;

public class InterfaceGrafica extends JFrame{
```

```
        //objetos da tela
        private JLabel lblCodigo;
        private JLabel lblNome;
        private JTextField txtNome;
        private JLabel lblIdade;
        private JTextField txtIdade;
        private JLabel lblRa;
        private JTextField txtRa;
        private JButton btnInserir;
        private JButton btnAtualizar;
        private JButton btnExcluir;
        private JTable tabela; //objeto novo
        final DefaultTableModel modelo; //objeto novo
        private Container tela;

        //public CRUD controle = new CRUD();//dará um erro, pois ainda não foi criada a classe CRUD. Pode deixar comentado nesse momento, mas depois descomentar.

        //configuração da tela no construtor
        public InterfaceGrafica()
        {
                setSize(400,500);
                setTitle("Cadastro de Aluno");
                tela = getContentPane();
                tela.setLayout(null);

                //configuração dos rótulos
                lblCodigo = new JLabel("Código:");
                lblNome = new JLabel("Nome:");
                lblIdade = new JLabel("Idade:");
                lblRa = new JLabel("RA:");

                lblCodigo.setBounds(10,10,80,25);
                lblNome.setBounds(10, 45,80, 25);
                lblIdade.setBounds(10,85,80,25);
                lblRa.setBounds(10,135,80,25);
                tela.add(lblCodigo);
                tela.add(lblNome);
                tela.add(lblIdade);
                tela.add(lblRa);

                //configuração das caixas de texto
                txtCodigo = new JTextField();
                txtNome = new JTextField();
                txtIdade = new JTextField();
                txtRa = new JTextField();

                txtCodigo.setBounds(90,10,40,25);
                txtNome.setBounds(90,45,150,25);
                txtIdade.setBounds(90,85,40,25);
                txtRa.setBounds(90,135,150,25);
                tela.add(txtCodigo);
                tela.add(txtNome);
                tela.add(txtIdade);
                tela.add(txtRa);

                //configuração dos botões
```

```
        btnInserir = new JButton("Inserir");
        btnAtualizar = new JButton("Atualizar");
        btnExcluir = new JButton("Excluir");

        btnInserir.setBounds(10,170,100,50);
        btnAtualizar.setBounds(110,170,100,50);
        btnExcluir.setBounds(210,170,100,50);
        tela.add(btnInserir);
        tela.add(btnAtualizar);
        tela.add(btnExcluir);

        //configuração da tabela
        modelo = new DefaultTableModel();
        tabela = new JTable(modelo);
        modelo.addColumn("Código");
        modelo.addColumn("Nome");
        modelo.addColumn("Idade");
        modelo.addColumn("RA");
        JScrollPane painel = new JScrollPane(tabela);
        painel.setBounds(10, 240, 300, 200);
        tela.add(painel);
        carregarDados();

        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //tratamento dos eventos de clique nos botões
        btnInserir.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                //preencher depois
            }
        });

        btnAtualizar.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                //preencher depois
            }
        });

        btnExcluir.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                //preencher depois
            }
        });
    }

    private void carregarDados() {
        //preencher depois
    }

    public static void main(String[] args) {
        InterfaceGrafica tela1 = new InterfaceGrafica();
    }
```

```
}
```

Agora vamos implementar o CRUD. Crie um banco de dados chamado de “cadastro” usando o MySql. Sugiro usar o MySql Workbench ou phpMyAdmim. Dentro do banco criado, crie uma tabela chamada “aluno” com os campos: id_aluno, nome, idade, ra. A interface gráfica do workbench pode ser usada para ganhar tempo, mas segue o código sql caso opte por digitar.

```
CREATE TABLE `cadastro`.`aluno` (
  `id_aluno` INT NOT NULL AUTO_INCREMENT,
  `nome` VARCHAR(45) NULL,
  `idade` INT NULL,
  `ra` VARCHAR(45) NULL,
  PRIMARY KEY (`id_aluno`));
```

Volte ao projeto criado anteriormente “CadastroAluno” e adicione o conector mysql ao classpath. Para isso faça o download do conector J se necessário, descompacte-o em uma pasta no computador. Clique com o botão direito no projeto do Eclipse, selecione Properties >> Java Build Path na aba Libraries, adicione um JAR externo.

Crie uma nova classe chamada “ClasseConexao”. Essa classe será responsável por estabelecer a conexão com o banco de dados, com o respectivo código:

```
import java.sql.*;

public class ClasseConexao {
	//parâmetros para conexão
	private Connection conexao;
	private String URLBD =
"jdbc:mysql://localhost:3306/cadastro?useSSL=false&serverTimezone=UTC";
	private String usuario="root";
	private String senha="1234";

	//criando a conexão no construtor
	public ClasseConexao() {
		try {
			Class.forName("com.mysql.cj.jdbc.Driver");
			conexao = DriverManager.getConnection(URLBD,usuario,senha);

		}catch(Exception ex) {
			ex.printStackTrace();
		}
	}

	//criando um método para acessar a conexão
	public Connection getConexao() {
		return conexao;
	}
}
```

Agora crie uma nova classe chamada “CRUD”, essa classe conterá as instruções para realizar as operações no banco de dados.

```
import java.sql.*;

public class CRUD {
	private ClasseConexao conexao;

	public CRUD() {
		conexao = new ClasseConexao();
	}
	//create
	public boolean cadastrarAluno(String nome, int idade, String ra) {
		try {
			PreparedStatement comando =
conexao.getConexao().prepareStatement("INSERT INTO aluno (nome,idade,ra) VALUES (?,?,?)");
			comando.setString(1,nome);
			comando.setInt(2,idade);
			comando.setString(3,ra);
			return comando.execute();
		} catch (Exception e) {
			e.printStackTrace();
			return false;
		}
	}

	//read
	public ResultSet carregarAlunos() {
		ResultSet dados = null;
		try {
			PreparedStatement comando =
conexao.getConexao().prepareStatement("SELECT * from aluno");
			dados = comando.executeQuery();
		} catch (Exception e) {
			e.printStackTrace();
		}
		return dados;
	}

	//update
	public boolean atualizarAluno(int codigo, String nome, int idade, String ra)
	{
		try {

			PreparedStatement comando =
conexao.getConexao().prepareStatement("UPDATE aluno SET nome=?,idade=?,ra=? WHERE
id_aluno=?");
			comando.setString(1,nome);
			comando.setInt(2,idade);
			comando.setString(3,ra);
			comando.setInt(4, codigo);
			return comando.execute();

		} catch (Exception e) {
```

```
                e.printStackTrace();
                return false;
            }
        }

        //delete
        public boolean excluirAluno(int codigo)
        {
            try {
                PreparedStatement comando =
conexao.getConexao().prepareStatement("DELETE FROM aluno WHERE id_aluno=?");
                comando.setInt(1,codigo);
                return comando.execute();

            } catch (Exception e) {
                e.printStackTrace();
                return false;
            }
        }
}
```

Com as classes devidamente implementadas agora é necessário voltar a classe "InterfaceGrafica" para completar o código que faltava. Localize os métodos de tratamaento aos eventos e complete com o código:

```
        //tratamento dos eventos de clique nos botões
            btnInserir.addActionListener(new ActionListener() {

                @Override
                public void actionPerformed(ActionEvent e) {
                    //preencher depois
                    controle.cadastrarAluno(txtNome.getText(),
Integer.parseInt(txtIdade.getText()), txtRa.getText());
                    carregarDados();
                }
            });

            btnAtualizar.addActionListener(new ActionListener() {

                @Override
                public void actionPerformed(ActionEvent e) {
                    //preencher depois
        controle.atualizarAluno(Integer.parseInt(txtCodigo.getText()), txtNome.getText(),
Integer.parseInt(txtIdade.getText()), txtRa.getText());
                    carregarDados();
                }
            });

            btnExcluir.addActionListener(new ActionListener() {

                @Override
                public void actionPerformed(ActionEvent e) {
                    //preencher depois
        controle.excluirAluno(Integer.parseInt(txtCodigo.getText()));
                    carregarDados();
```

```
                        }
                });
        }

        private void carregarDados() {
                //preencher depois
                modelo.setNumRows(0);
                try {
                        ResultSet dados = controle.carregarAlunos();
                        while(dados.next()){
                                int id = dados.getInt("id_aluno");
                           String nome = dados.getString("nome");
                           int idade = dados.getInt("idade");
                           String ra = dados.getString("ra");
                           modelo.addRow(new Object[]{new Integer(id), nome, new
Integer(idade),ra});
                         }
                }catch(Exception ex) {
                        ex.printStackTrace();
                }
        }

        public static void main(String[] args) {
                InterfaceGrafica tela1 = new InterfaceGrafica();
        }
```

Nesse momento tudo deverá estar devidamente funcionando.

Checklist:

Verificar o nome da classe;

Verificar se todos os objetos foram criados;

Verificar se o construtor da classe foi devidamente codificado;

Verificar se os métodos que irão tratar os eventos foram em seus devidos lugares para receber o código na próxima atividade.

Criar um novo banco de dados MySql chamado “cadastro” usando um SGBD;

Criar a tabela cliente, com os campos, código, nome, idade e registro acadêmico;

Criar uma classe para a conesão do banco de dados “ClasseConexao”;

Criar uma classe para as operações do banco. “CRUD”

Voltar na classe de interface gráfica e inserir os códigos para tratamento dos eventos.

RESULTADOS

Resultados da aula prática:

Ao final da aula prática deverá ser entregue um projeto que contenha uma *interface* gráfica, bem como a implementação da persistência de dados usando o MySQL.