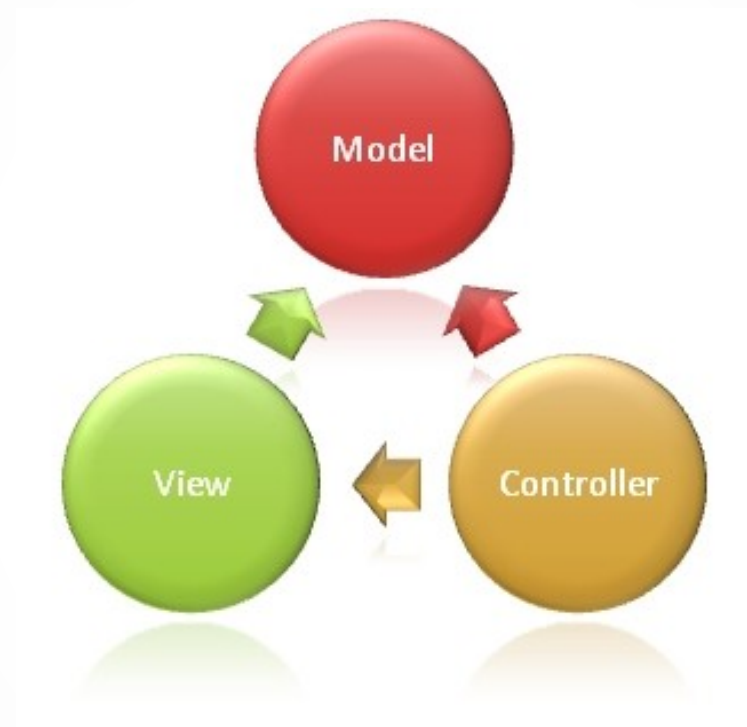


Java MVC



Prof. Dr. João Paulo Lemos Escola
Copyright© 2022

Tópicos da aula

- Nesta aula vamos começar a estudar o modelo MVC com Java;
- Vamos entender as camadas MVC;
- Aprender como utilizar cada camada.

Camadas MVC

- Model
 - Modelo;
 - Classes que representam as entidades do banco de dados;
- View
 - Visão;
 - Classes responsáveis por apresentar as interfaces para o usuário;
- Controller
 - Controlador;
 - Classes com as regras de negócio;
- Dao (Data Access Objects)
 - Objetos de acesso a dados;
 - Classes responsáveis por acessar o banco de dados.

Pacote Controller

- Este pacote armazena as classes com as regras de negócio;
- Estas classes são intermediárias entre as Views e as DAOs;
- Tudo que for feito na View deve empregar um método de uma classe Controller, que por sua vez irá executar processamentos ou armazenamentos utilizando (ou não) as DAOs.

Pacote DAO

- O pacote DAO armazena as classes de acesso a dados (Data Access Objects);
- Teremos aqui a classe que vai ser responsável pela comunicação com o banco de dados e as classes que vão ter os métodos básicos de manipulação de dados: inserir, excluir, alterar e buscar por exemplo.

Pacote Model

- Neste pacote teremos as classes que representam os objetos do mundo real;
- Por exemplo: “Pessoa” com seus campos (provavelmente os mesmos que foram modelados no BD) e os métodos get e set;

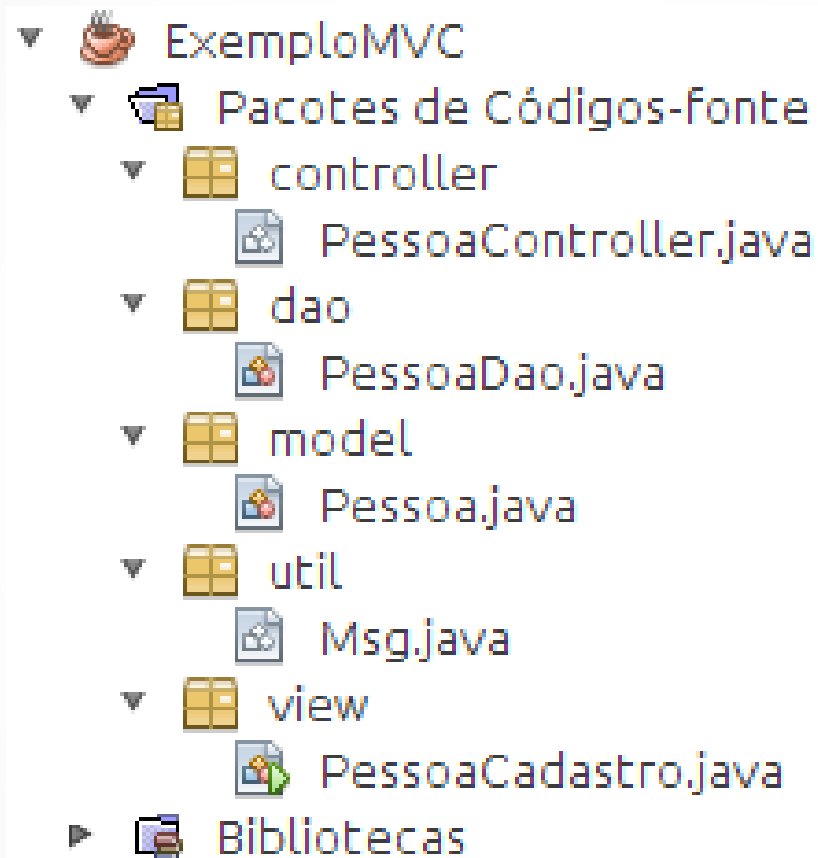
Pacote View

- Neste pacote teremos as classes de janelas gráficas a serem apresentadas ao usuário;
- Vamos utilizar o padrão PessoaListagem, ProdutoCadastro etc.

Pacote Util

- Não é obrigatório;
- Normalmente utilizado para guardar as classes com métodos auxiliares;
- Exemplos:
 - Geração de pdf;
 - Encapsulamento de tarefas (alertas, ordenação);
 - Métodos auxiliares diversos.

Estrutura das classes e pacotes



Pessoa.java

```
package model;

/**
 *
 * @author jpescola
 */
public class Pessoa {
    private int codigo;
    private String nome;
    private String cpf;

    public Pessoa() {
    }

    public Pessoa(int codigo, String nome, String cpf) {
        this.codigo = codigo;
        this.nome = nome;
        this.cpf = cpf;
    }

    public int getCodigo() {
        return codigo;
    }

    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }
}
```

PessoaController.java

```
package controller;

import dao.PessoaDao;
import model.Pessoa;

/**
 *
 * @author jpescola
 */
public abstract class PessoaController {

    public static boolean cadastrar(Pessoa p){
        PessoaDao pessoaDao = new PessoaDao();
        return pessoaDao.inserir(p);
    }

}
```

PessoaDao.java

```
import java.util.List;
import model.Pessoa;

/**
 *
 * @author jpescola
 */
public class PessoaDao {

    public boolean inserir(Pessoa p){
        return true;
    }

    public boolean excluir(Pessoa p){
        return true;
    }

    public boolean alterar(Pessoa p){
        return true;
    }

    public List<Pessoa> buscar(Pessoa p){
        return new ArrayList<Pessoa>();
    }

}
```

PessoaCadastro.java

```
public class PessoaCadastro {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
  
        int codigo = Integer.parseInt(Msg.input("Codigo:"));  
        String nome = Msg.input("Nome:");  
        String cpf = Msg.input("CPF:");  
  
        Pessoa pessoa1 = new Pessoa(codigo,nome,cpf);  
  
        if (PessoaController.cadastrar(pessoa1))  
            Msg.info("Sucesso ao cadastrar!");  
        else  
            Msg.erro("Erro ao cadastrar!");  
    }  
}
```

Msg.java

```
package util;

import javax.swing.JOptionPane;

/**
 *
 * @author jpescola
 */
public abstract class Msg {

    private static String titulo = "Sistema do JP";

    public static String input(String texto){
        return JOptionPane.showInputDialog(null,
            texto,
            titulo,
            JOptionPane.QUESTION_MESSAGE
        );
    }

    public static void info(String texto) {
        JOptionPane.showMessageDialog(null, texto, titulo, JOptionPane.INFORMATION_MESSAGE);
    }

    public static void erro(String texto) {
        JOptionPane.showMessageDialog(null, texto, titulo, JOptionPane.ERROR_MESSAGE);
    }
}
```

A10ex1.jar

- Crie um projeto MVC de cadastro de produtos a partir das classes exemplo da aula atual.

A10ex2.jar

- Crie um projeto de cadastro de Usuários com herança na classe Pessoa.

O que aprendemos?

- Nesta aula estudamos o modelo MVC com Java;
- Camadas MVC.

Na próxima aula...

- Vamos aprender a acessar um servidor de banco de dados a partir de uma classe Java;
- Utilizar o SGBD MySQL instalado na máquina atual, utilizando o WampServer;
- As classes de banco de dados serão criadas no pacote DAO.