

Padrão MVC e Observer com Java

Departamento Acadêmico de Ciência da Computação (DACC)

Profº Wellington Moreira de Oliveira

Padrão MVC (Model View Controller)

- ▶ Model
 - ▶ dados ou lógica de negócios (domínio)
 - ▶ Podemos ter validações, acesso a banco, acesso à arquivos, cálculos, etc.
- ▶ View
 - ▶ Apresentação dos dados (interface).
 - ▶ Meio pelo qual o usuário pode acessar e mudar o modelo
- ▶ Controller
 - ▶ Gerenciamento do fluxo da aplicação
 - ▶ Transforma eventos gerados pela interface em ações de negócio, alterando o modelo
 - ▶ Conhece o Model e a View

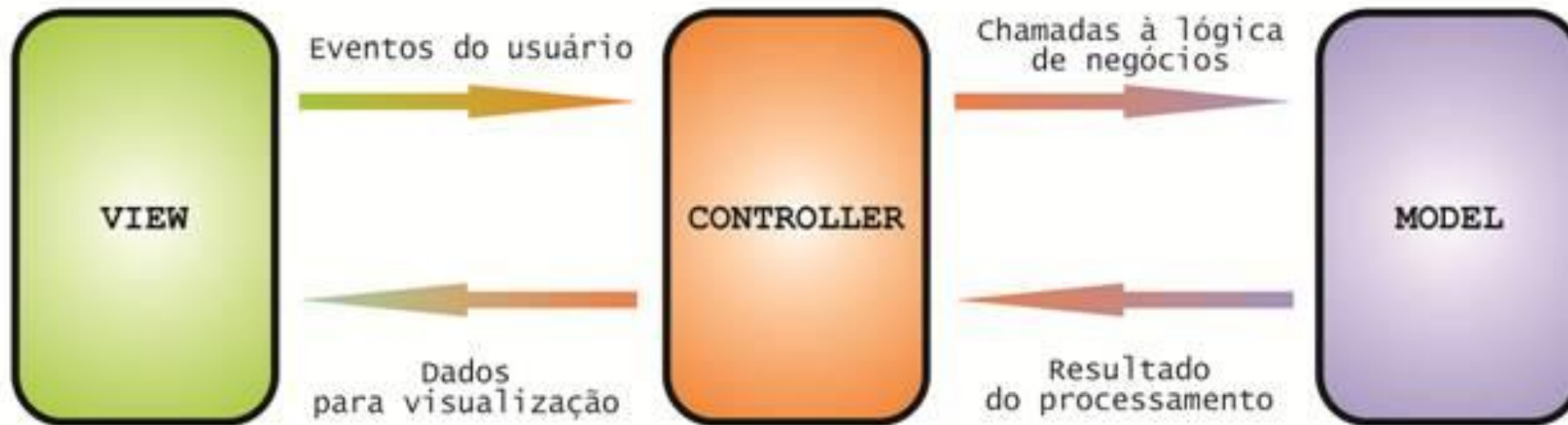
Por que usar MVC?

- ▶ Mudanças constantes do código fonte
- ▶ Flexibilidade para a utilização em outras plataformas
 - ▶ Web, mobile, etc
- ▶ Atender diferentes requisitos de interface fica mais fácil
 - ▶ visões diferentes sobre os mesmos dados e lógica
- ▶ Independência
 - ▶ A lógica de negócio não deve conhecer as interfaces que exibem seu estado

Como usar MVC?

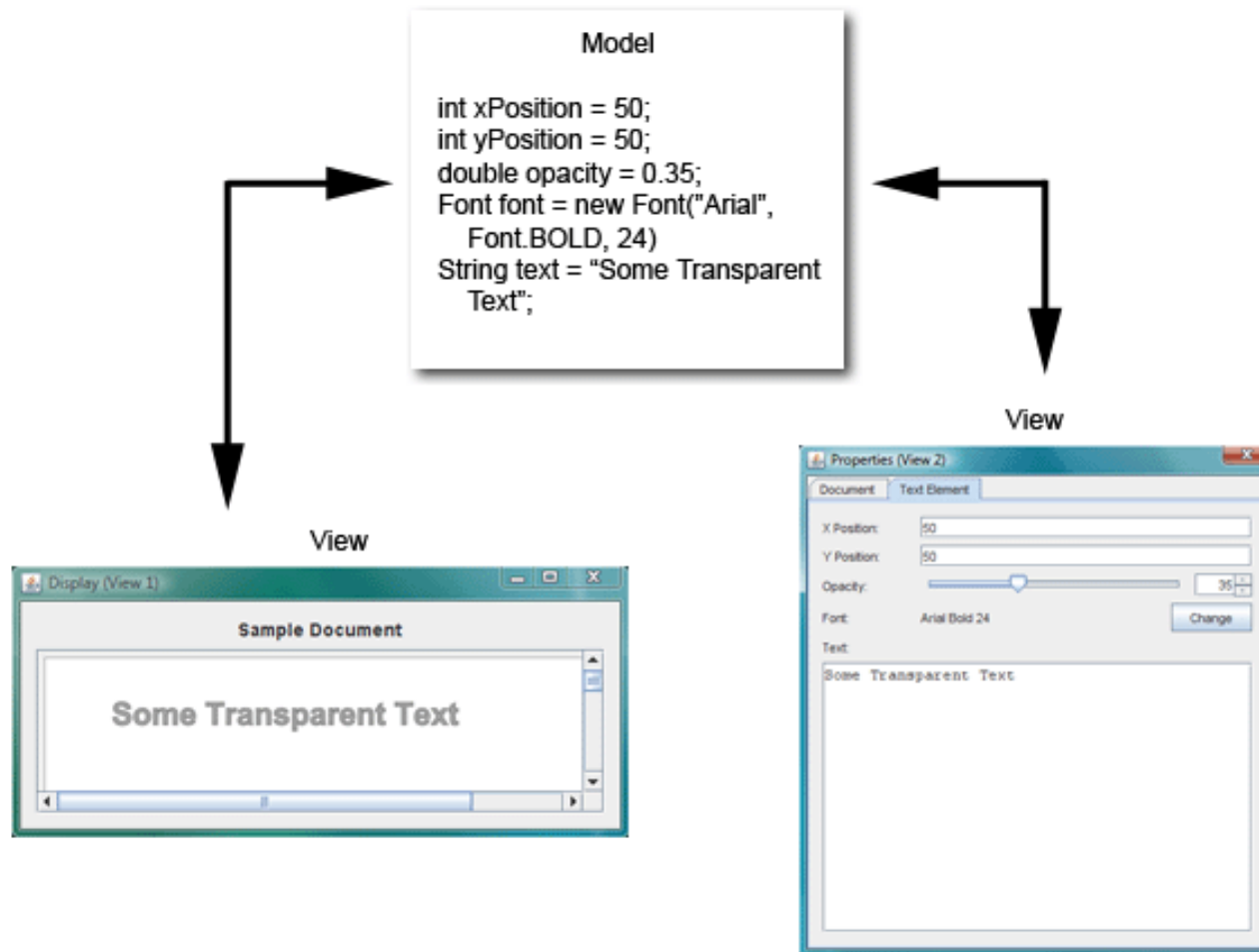
- ▶ Crie as classes de modelo de acordo com seu diagrama de classes
- ▶ Crie um pacote **Model** para isolar as classes de domínio
- ▶ Crie um pacote **View** para isolar as classes de interface (Ex.: Frames do Swing)
- ▶ Crie um pacote **Controller** para isolar as classes de gerenciamento de alterações da interface sobre o model
- ▶ Isole toda codificação utilizando o Controller como Mediador para que o Model não conheça a View e vice-e-versa

Controlador como mediador



Fonte: <https://www.devmedia.com.br/padrao-mvc-java-magazine/21995>

Duas visões do mesmo modelo



Referências

- ▶ <http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/arqu/mvc/mvc.htm>
- ▶ <http://www.oracle.com/technetwork/articles/javase/index-142890.html>
- ▶ <http://www.klebermota.eti.br/2011/12/23/desenvolvimento-de-uma-aplicacao-java-se-com-o-mvc/>
- ▶ <https://www.devmedia.com.br/padrao-mvc-java-magazine/21995>

Padrão MVC e Observer com Java

Departamento Acadêmico de Ciência da Computação (DACC)

Profº Wellington Moreira de Oliveira