Computação II – Orientação a Objetos

Fabio Mascarenhas - 2016.2

http://www.dcc.ufrj.br/~fabiom/java

Interfaces Gráficas

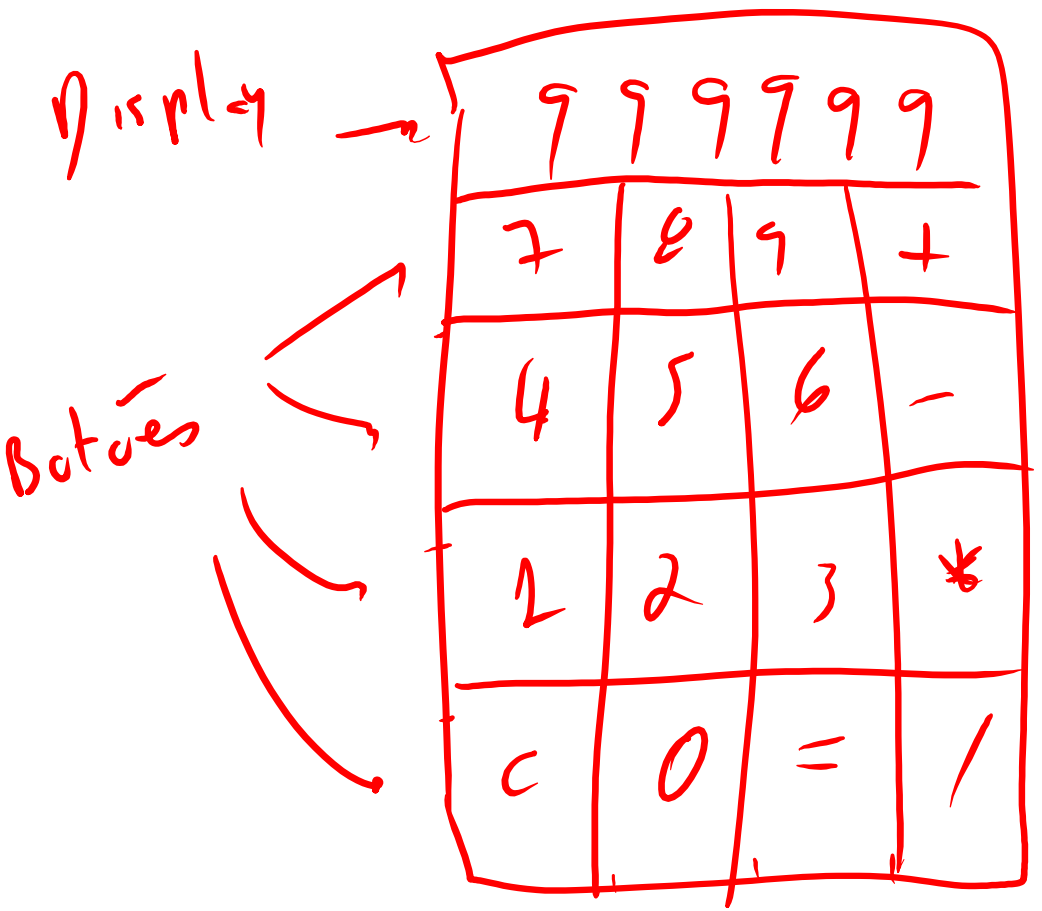
- Vamos usar nosso framework do Motor, com pequenas mudanças (para permitir interação com o mouse) para implementar não um jogo, mas uma aplicação com uma pequena interface gráfica

- Vamos fazer dois programas: uma calculadora de quatro funções, e um editor de figuras geométricas

- As duas aplicações vão ter vários *controles:* botões de comando, caixas de texto, áreas de desenho, sliders

- Vamos ter também undo e redo (desfazer e refazer) de vários níveis

Calculadora – esboço da interface



Model-View-Controller (MVC)

- O MVC é o principal padrão para estruturação de aplicações com interfaces gráficas

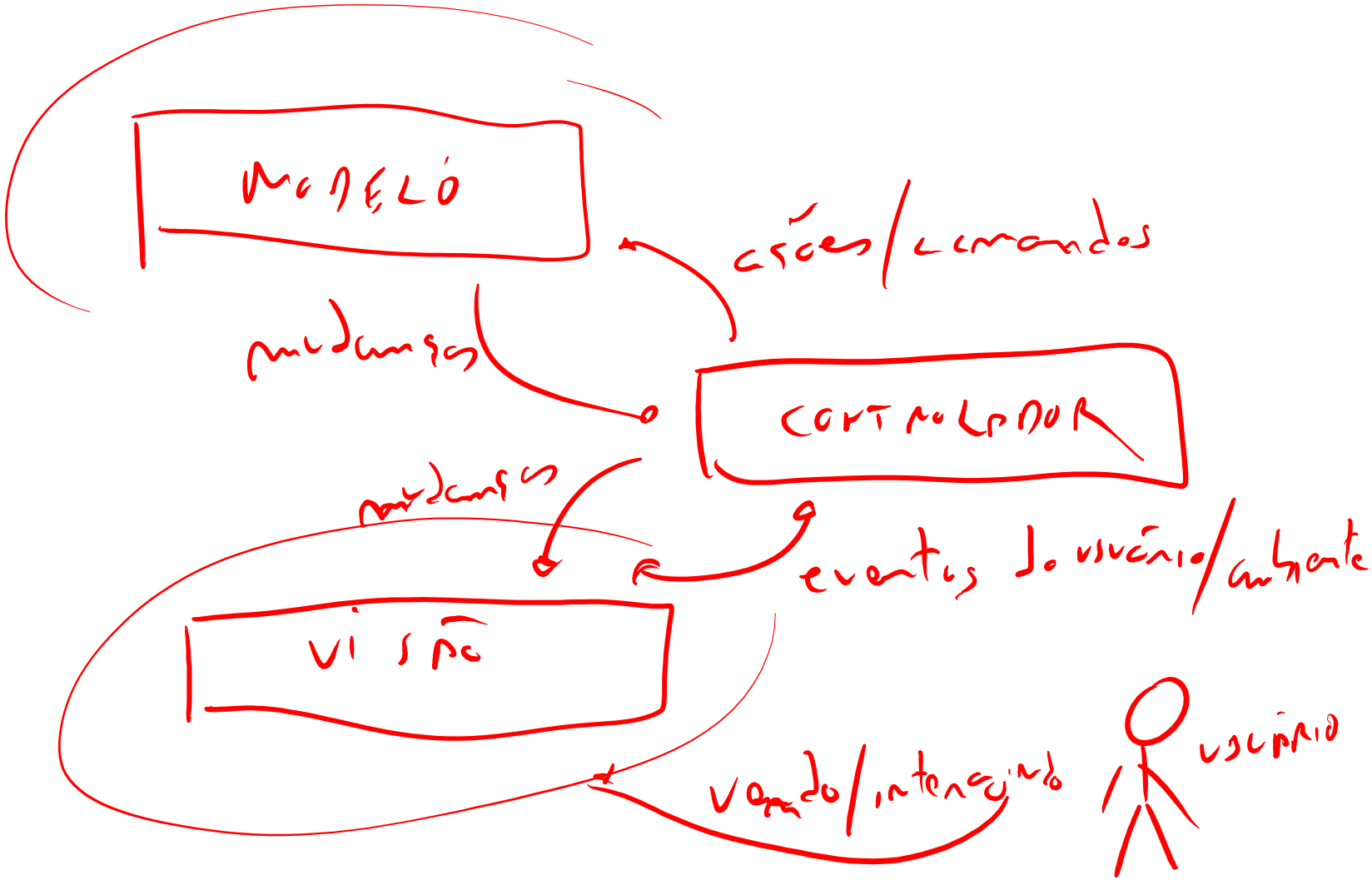
- Ele separa a aplicação em três grandes partes:

  - O *modelo* representa os dados da aplicação, e implementa a sua lógica interna de uma maneira o mais independente dos detalhes da interface quanto possível

  - A *visão* é a parte visível da interface, e é como o usuário enxerga os dados do modelo

  - O *controlador* faz a mediação entre o usuário, o modelo e a visão

MVC em um diagrama



Componentes (visão) ou Controles

- A interface `Componente` é formada por quatro métodos `getX()`, `getY()`, `getLargura()` e `getAltura()`, que dão as *bordas* do componente

- Usamos essas coordenadas para saber se um clique do mouse está dentro do componente ou não

- Também temos um método `desenhar(Tela t)`, para o componente se desenhar

- Finalmente, temos quatro métodos, `clique`, `aperto`, `solta` e `arrasto`, que dão os eventos do mouse

- Cada um desses métodos recebe as coordenadas `x` e `y` do evento

Botões

- Um botão é nosso componente mais simples
- Quando um botão é clicado, ele deve executar uma *ação*: essa ação é instância de uma interface `Acao` bem simples, com um único método `executar` sem parâmetros
- O botão também monitora quando o mouse é apertado ou solto, para fornecer *feedback* visual ao usuário de que ele está sendo clicado
- O controlador despacha cliques do mouse para o botão apropriado

O modelo da calculadora

- O modelo da calculadora é uma instância de `ModeloCalc`, mais todo o seu estado interno

- Esse estado consiste no valor do display, na *operação pendente*, e na última operação feita, e o observador do modelo

- Essa operação é uma instância da interface `OpCalc`; a própria classe `ModeloCalc` deriva de uma interface `Calc`, para podermos ter outros modos de operação

```
public interface OpCalc {
  void setEsq(int x);
  void setDir(int x);
  int valorOp();
}
```

```
public interface Calc {
  void digito(int n);
  void soma();
  void sub();
  void mult();
  void div();
  void igual();
  void reset();
  void setObservador(ObservadorDisplay obs);
}
```

MVC na calculadora

- Em nossa calculadora as visões serão os botões e a caixa de texto que mostra o resultado

- Todas as visões são instâncias da interface `Componente`, que é como elas interagem com o controlador

- Essas outras interfaces são *observadores* do modelo

- O controlador é a classe `Calculadora`, uma instância da interface `App`, assim como as instâncias da interface `Acao` (anônimas ou não) ligadas a cada botão

Observadores ou listeners

- Usamos o padrão observador quando queremos desacoplar objetos que *emitem* eventos dos objetos que os consomem

- A fonte ou sujeito dos eventos pode implementar uma interface que permite aos observadores se cadastrar para receber eventos, e se descadastrar, mas também é comum ter sujeitos com um único observador

- Os observadores ou *listeners* implementam uma interface que permite que sejam notificados caso algum evento ocorra

- É um padrão muito usado em interfaces gráficas

Exemplos de observadores

- Já usamos muitos observadores:
  - A interface `Jogo` é em parte um observador, assim como as interfaces `App` e `Componente`
  - As interfaces `*Listener` usadas em `Motor` para despachar os eventos para a aplicação
  - A interface `Acao` para conectar um botão ao que fazer quando é clicado
  - A interface `ObservadorDisplay` para conectar o valor do display da calculadora à caixa de texto