DCC / ICEx / UFMG

Padrões de Projeto Comportamentais

Eduardo Figueiredo

http://www.dcc.ufmg.br/~figueiredo

Padrões Comportamentais

- Chain of Responsibility (CoR)
- Command
- Interpreter
- Iterator
- Mediator
- Memento
- Observer
- State
- Strategy
- Template Method
- Visitor

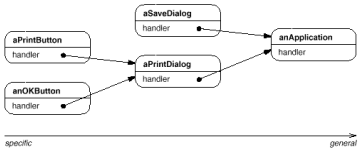
Chain of Responsibility (CoR)

Motivação

- Há vários objetos que possivelmente podem tratar uma determinada solicitação
  - O remetente da solicitação (cliente) não sabe qual objeto sabe tratar a solicitação
- Assim, a solicitação precisa passar por vários objetos até encontrar o tratamento adequado

Exemplo de Aplicação

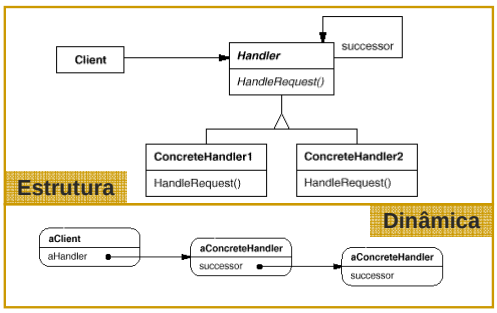
- Em uma aplicativo com ajuda sensível a contexto, ao pedir ajuda sobre um botão:
  - O próprio botão pode saber explicar sua função
  - Ou a janela pode explicar a função do botão
  - Ou talvez a ajuda pode estar na aplicação



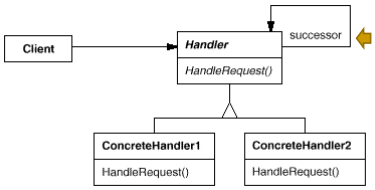
Padrão Chain of Responsibility

- Nome
  - Chain of Responsibility (CoR)
- Descrição do problema
  - Evitar acoplamento direto entre o remetente de uma solicitação e o receptor / tratador
- Descrição da solução (próximo slide)
- Consequências
  - Desacoplar remetente de receptor
  - Flexibilidade para incluir receptores

Solução do CoR

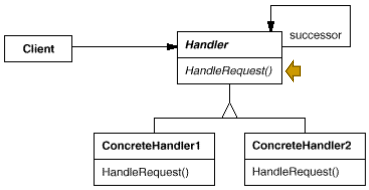


Sucessor na Cadeia



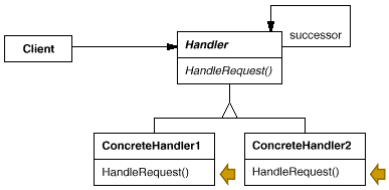
Todo tratador (*handler*) possui uma referência para o seu sucessor na cadeia.

A Classe Handler



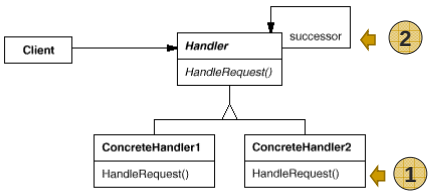
A classe *Handler* (abstrata) possui a assinatura do método de tratamento.

As Classes ConcreteHandler



Cada classe concreta (*ConcreteHandler*) implementa um tratamento específico.

Opções de Tratamento



Um objeto tratador tem duas opções:
1. **Tratar a solicitação; ou**
2. **Passar a solicitação para o próximo tratador.**

Mediator

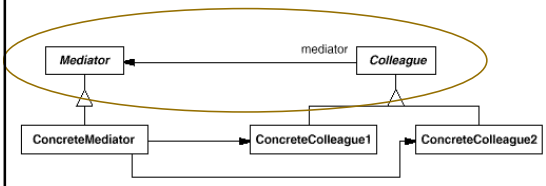
Motivação

- Considere a janela de uma aplicação
  - Existe dependências entre os componentes de interface (*widgets*)
  - Eles precisam se comunicar
- Por exemplo, um botão se torna inativo quando um campo está vazio

Padrão Mediator

- Nome
  - Mediator
- Descrição do problema
  - Evitar que objetos se relacionem diretamente. Reduzir o acoplamento.
- Descrição da solução (próximo slide)
- Consequências
  - Desacoplamento de objetos “colegas”
  - Centraliza o controle em um objeto

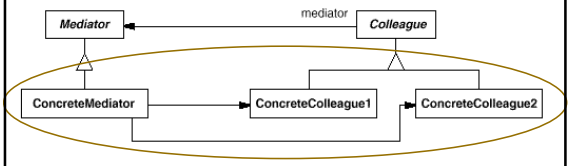
Solução Mediator



A classe *Colleague* conhece a classe *Mediator*.

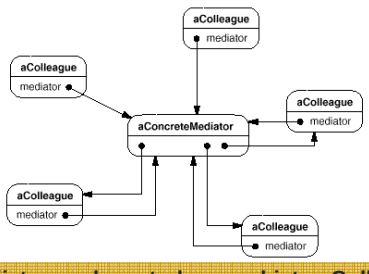
Os *ConcreteColleagues* se comunicaram com o *Mediator* por este relacionamento.

Solução Mediator



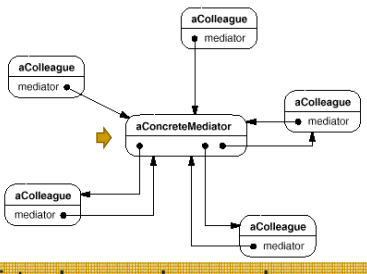
O *Mediator* (concreto) possui relacionamentos para todas as classes *ConcreteColleague*.

Dinâmica de Colaboração



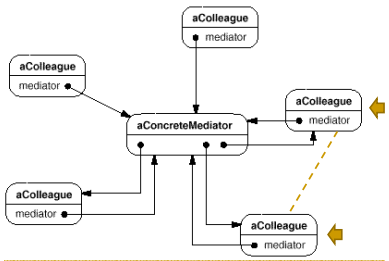
O *Mediator* conhece todos os objetos Colleague. Todos os *Colleagues* conhecem o mediator.

O Papel do Mediador



O Mediator desempenha o papel coorporativo de distribuição de solicitações ao objetos colegas.

Comunicação entre Colegas



Os objetos colegas se comunicam por meio do mediador.

State

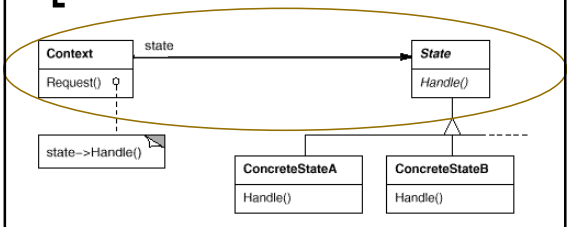
Motivação

- Considere um sistema de gerência e alocação de tarefas
  - Uma tarefa pode ter vários estados, como não iniciada, em andamento, concluída, cancelada, etc.
- O comportamento de uma tarefa depende do estado desta tarefa
  - Exemplo: Uma tarefa cancelada não aparece alocada a nenhuma pessoa

Padrão State

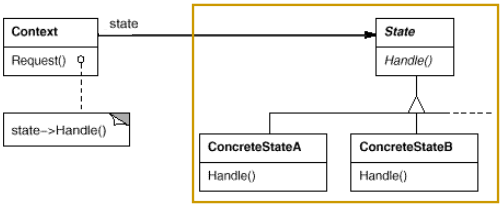
- Nome
  - State
- Descrição do problema
  - Alterar o comportamento de um objeto dependendo do seu estado
- Descrição da solução (próximo slide)
- Consequências
  - Torna explícito os estados de um objeto e as transições de estados

Solução State



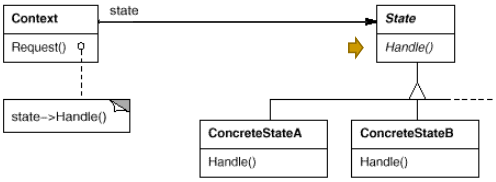
O contexto (*Context*) têm um estado ativo.
O contexto acessa o seu estado pela associação.

Estados Concretos



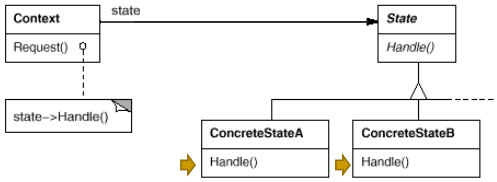
Os vários estados possíveis são implementados em classes concretas *ConcreteState* que estendem a classe abstrata *State*.

Classe State



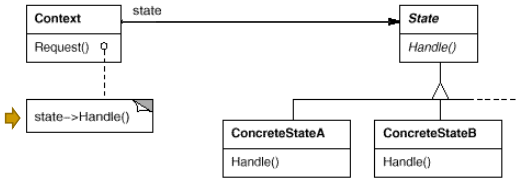
A classe abstrata *State* define os métodos comuns aos estados.

Classes ConcreteState



Cada estado concreto re-implementa os métodos da classe *State* dando o comportamento específico ao contexto.

Comportamento Específico



O comportamento específico do contexto é obtido chamando os métodos do estado atual.

Bibliografia da Aula

- E. Gamma, R. Helm, R. Johnson, J. Vlissides. **Padrões de Projeto**, 1a. Edição. Bookman, 2000.
  - Padrões: Chain of Responsibility, Mediator e State