PADRÕES DE PROJETO

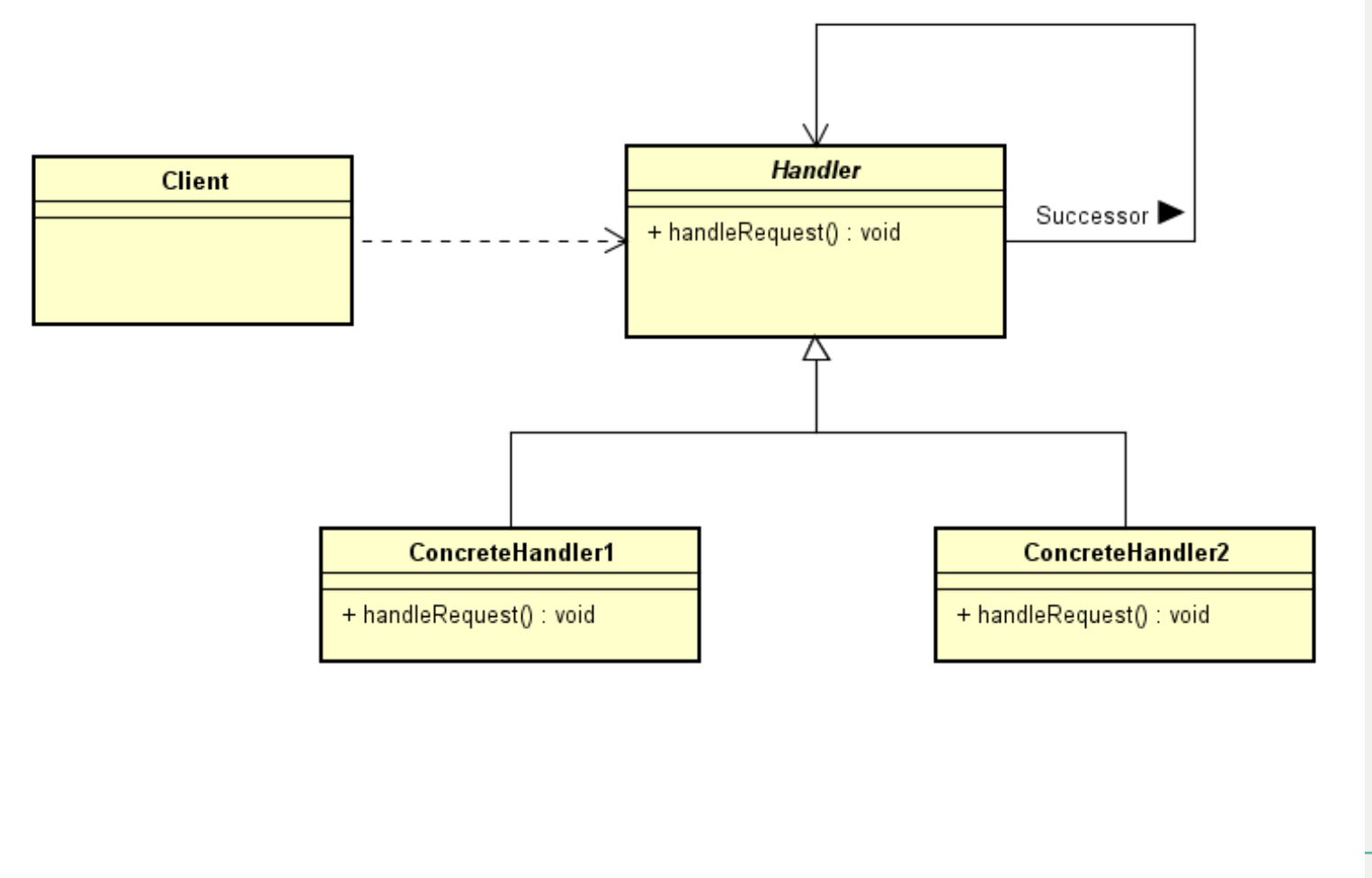
CHAIN OF RESPONSIBILITY

APRESENTAÇÃO: ALEXANDRE BEZERRA BARBOSA

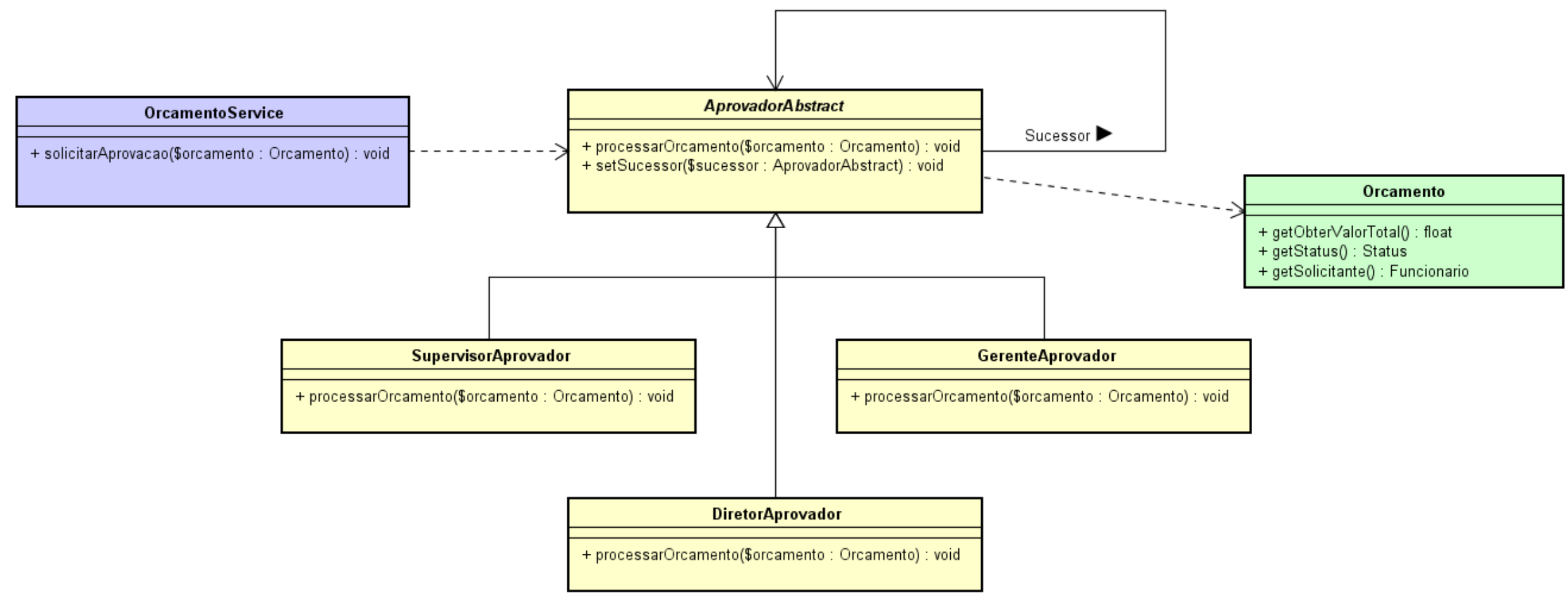
Chain of Responsibility

- Padrão comportamental
- Evita acoplamento entre remetente da solicitação e receptor da solicitação
- Cria uma estrutura encadeada por diferente Objetos Manipuladores
- Os Objetos Manipuladores podem decidir processar ou não a Requisição/Solicitação
- Mais de um manipulador poderá processar a solicitação
- Opera estruturado com composição de Objetos
- Permite uma estrutura com hierarquia e especializações de manipuladores em árvore com uso do padrão estrutural Composite.

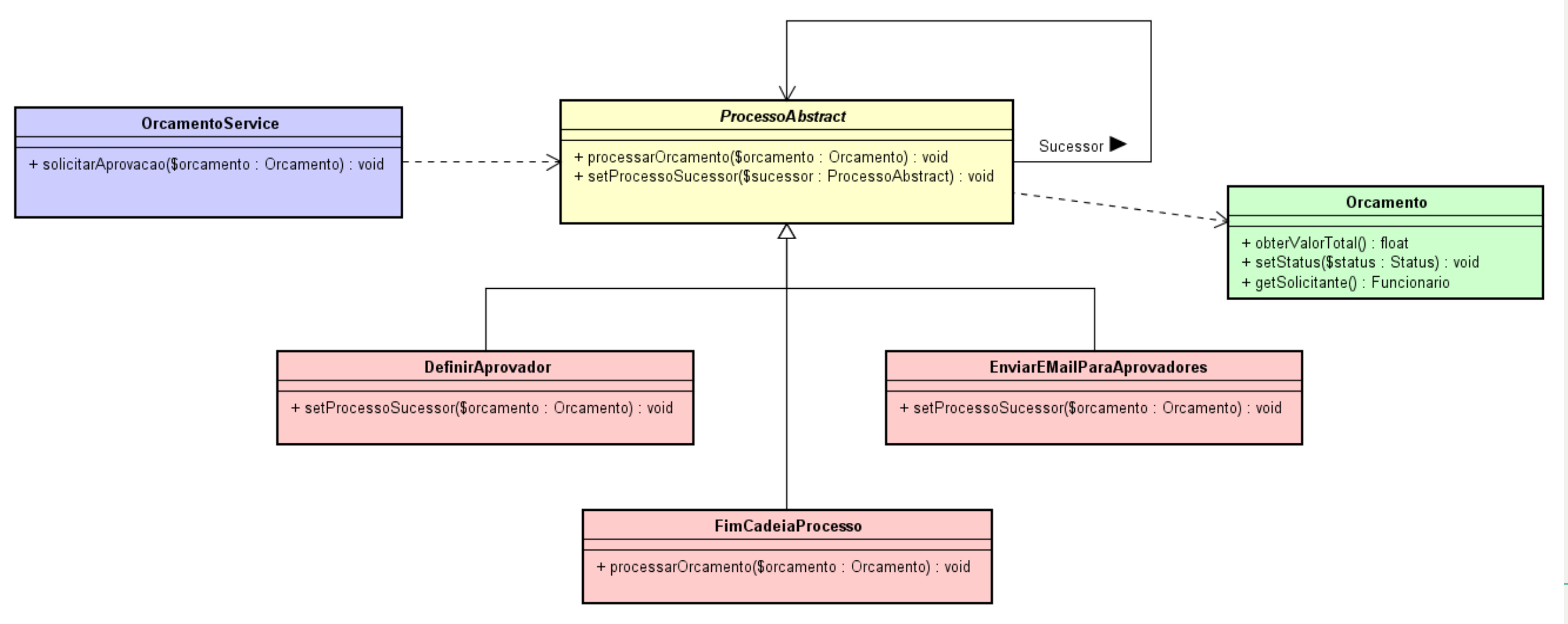
Quem são os elementos participantes?



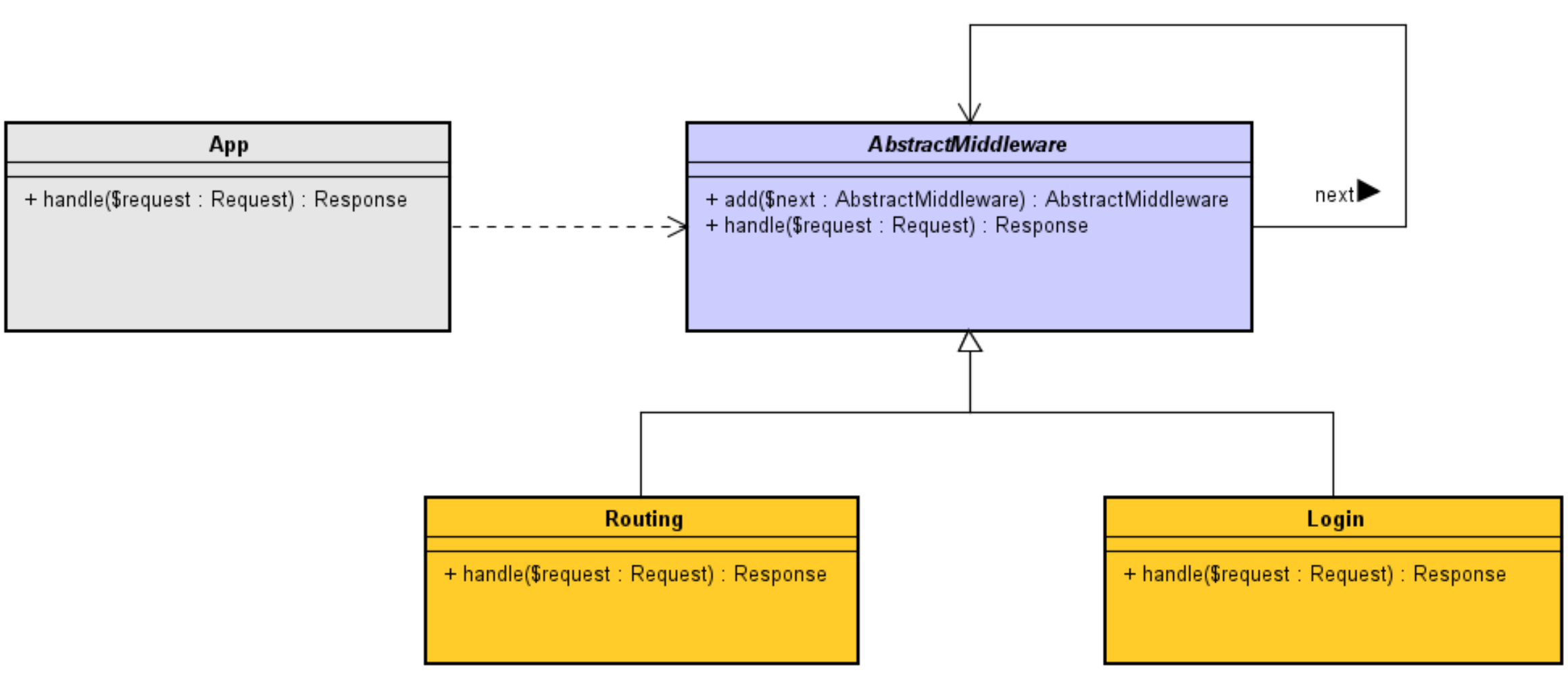
Exemplo de aplicação:



Exemplo de aplicação:



Exemplo de aplicação:



Abstract Handler

```php
abstract class AprovadorAbstract
{
    protected ?AprovadorAbstract $sucessor = null;

    public function setSucessor(AprovadorAbstract $sucessor): AprovadorAbstract
    {
        $this->sucessor = $sucessor;
        return $sucessor;
    }

    public abstract function processarOrcamento(Orcamento $orcamento): void;
}
```

Concrete Handler

```php
class SupervisorAprovador extends AprovadorAbstract
{

    public function processarOrcamento(Orcamento $orcamento): void
    {
        /** Poderá encerrar o processamento em cadeia */
        if ($orcamento->getStatus() !== Status::ABERTO) {
            return;
        }

       /** Decide se irá passar para classe sucessora */
        if ($orcamento->obterValorTotal() > NivelAprovacao::PRESIDENTE->value
            || ($orcamento->getAprovadores()->count() > 0)) {

            /** Poderá encerrar o processamento em cadeia */
            if (! $this->sucessor) {
                return;
            }
            /** Passa para a classe sucessora (Próximo handler) */
            $this->sucessor->processarOrcamento($orcamento);
        }

        // Implementação do processamento efetuado pelo handler...
    }
}
```

Cliente: implementação da chamada

Corrente de responsabilidade com manipuladores todos vinculados

```php
/** Implementação no cliente */

    $aprovadores = new SupervisorAprovador();
    $aprovadores
        ->setSucessor(new GerenteAprovador())
        ->setSucessor(new DiretorAprovador())
        ->setSucessor(new PresidenteAprovador())
        ->setSucessor(new ValorImprocedenteReprovar());

    $aprovadores->processarOrcamento($orcamento);


 /** Continuação do processo */
```

Envio da Solicitação/Requisição pelo remetente/Cliente