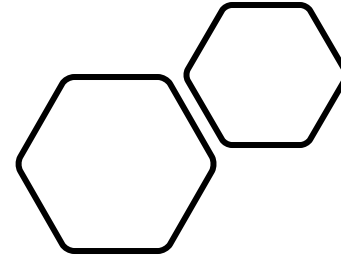


Arquitetura de Software

Prof. Bernardo Copstein



Os princípios SOLID parte 2

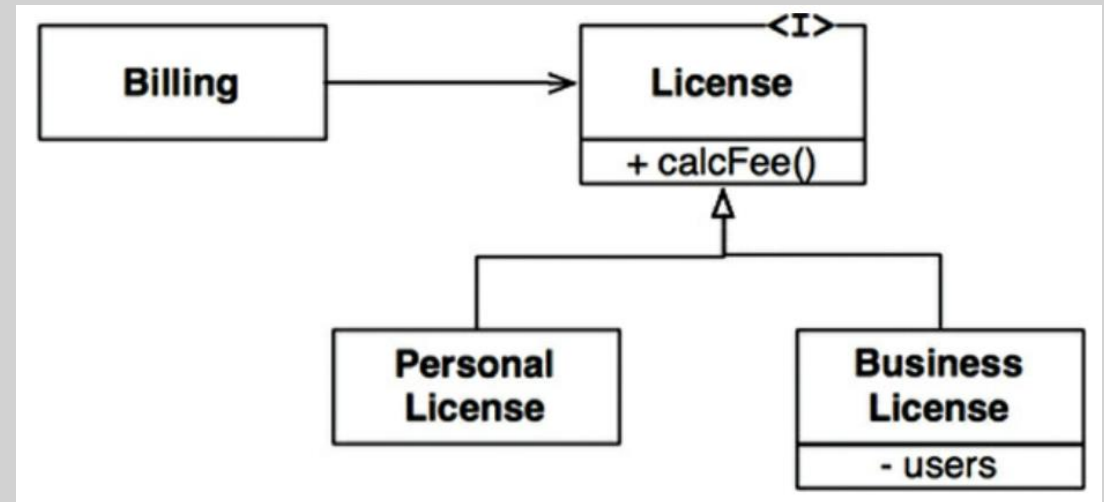


Leituras recomendadas:

- Martin, Robert C.. Clean Architecture (Robert C. Martin Series) . Pearson Education.

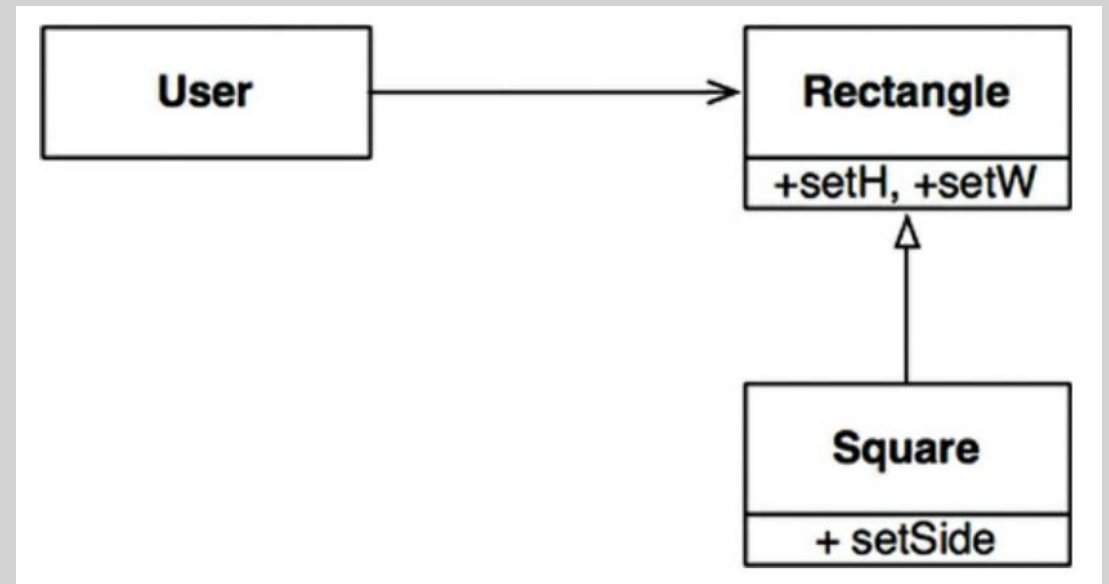
LSP: The Liskov Substitution Principle

- Se para cada objeto O1 do tipo S existe um objeto O2 do tipo T tal que para qualquer programa P definido em termos de T se o comportamento de P não muda quando O1 é substituído por O2, **então** S é subtipo de T.



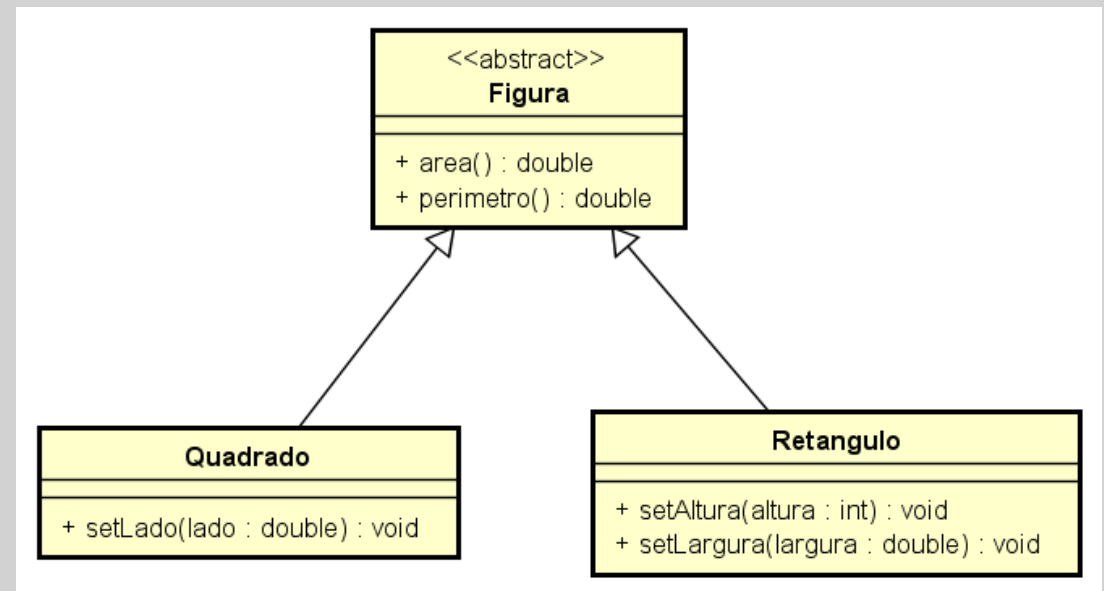
LSP: contra exemplo

- Os métodos setH e setW permitem que, eventualmente, uma instancia de “Square” termine por armazenar um retângulo.
- Não atende o **LSP**



LSP: neste caso o princípio é atendido

- Tanto *Retangulo* como *Quadrado* são figuras e o significado de área e perímetro são os mesmos nos dois casos.

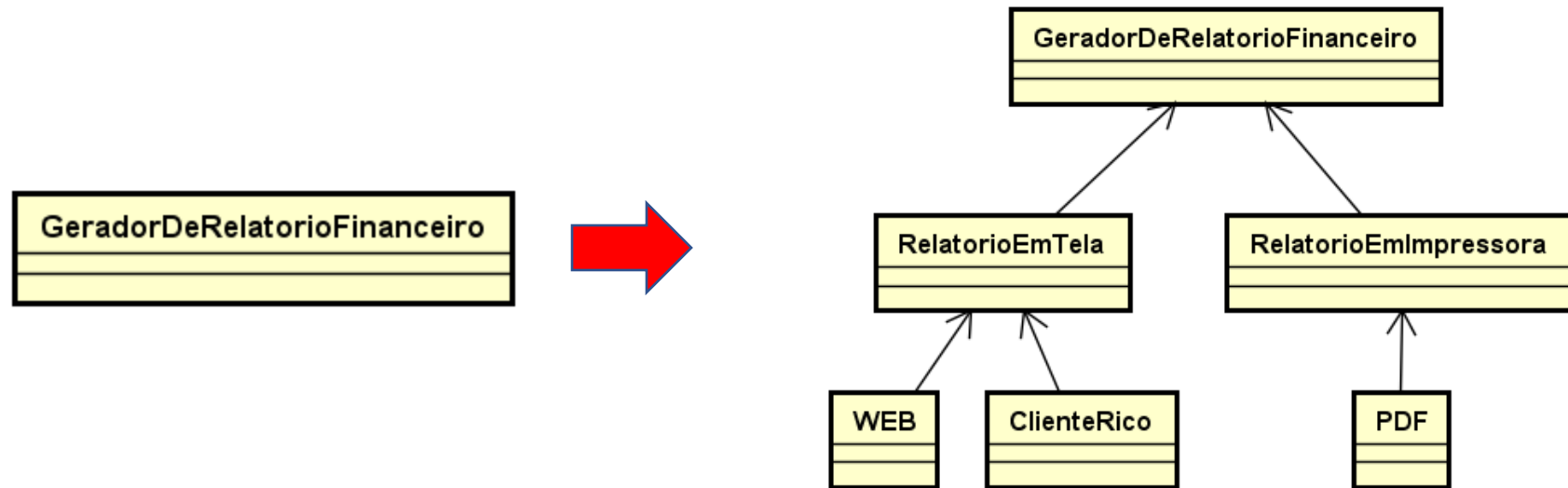


OCP: The Open- Closed Principle

“Um artefato de software deve ser aberto para extensões mas fechado para modificações”

- Deve ser possível expandir um módulo sem modificar o que já está pronto
- Como? Separando adequadamente as coisas que podem mudar por razões diferentes (SRP) e organizando as dependências entre elas adequadamente (DIP)

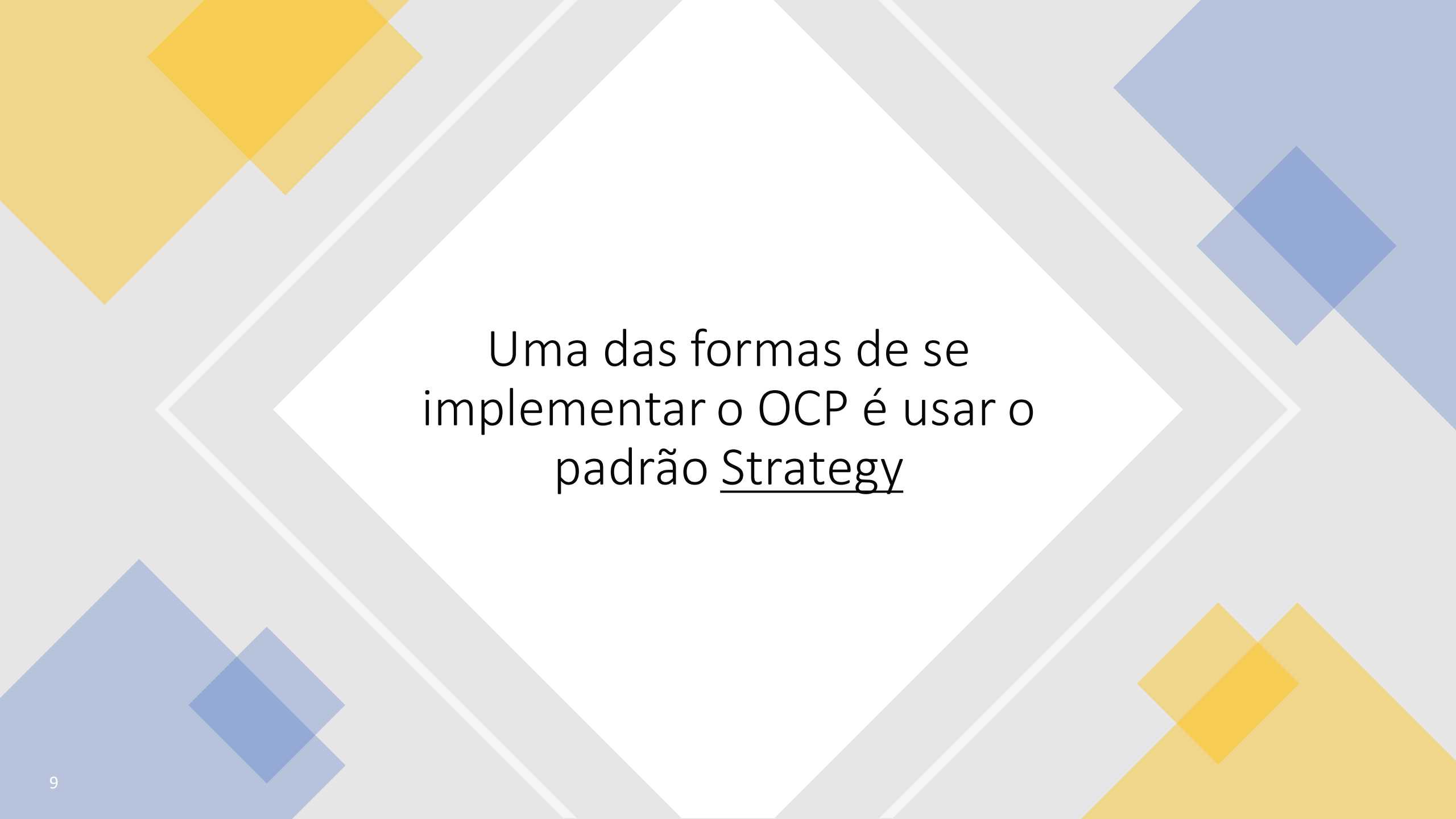
OCP: exemplo



As setas apontam na direção do módulo que se deseja proteger

OCP: resumindo

- O OCP é um dos norteadores da arquitetura de sistemas. O objetivo é fazer o sistema fácil de estender sem que sejam necessárias muitas alterações.
- O objetivo é atingido quebrando-se o sistema em componentes e organizando estes componentes em uma hierarquia de dependências que protege os componentes de mais alto nível de mudanças nos demais.



Uma das formas de se
implementar o OCP é usar o
padrão Strategy

Padrão Strategy

- Define uma família de algoritmos, encapsula cada um e permite que os mesmos sejam intercambiáveis. O padrão Strategy permite que o algoritmo varie independente dos clientes que o usam.

Padrão Strategy: Definição

Classificação: comportamental de objeto

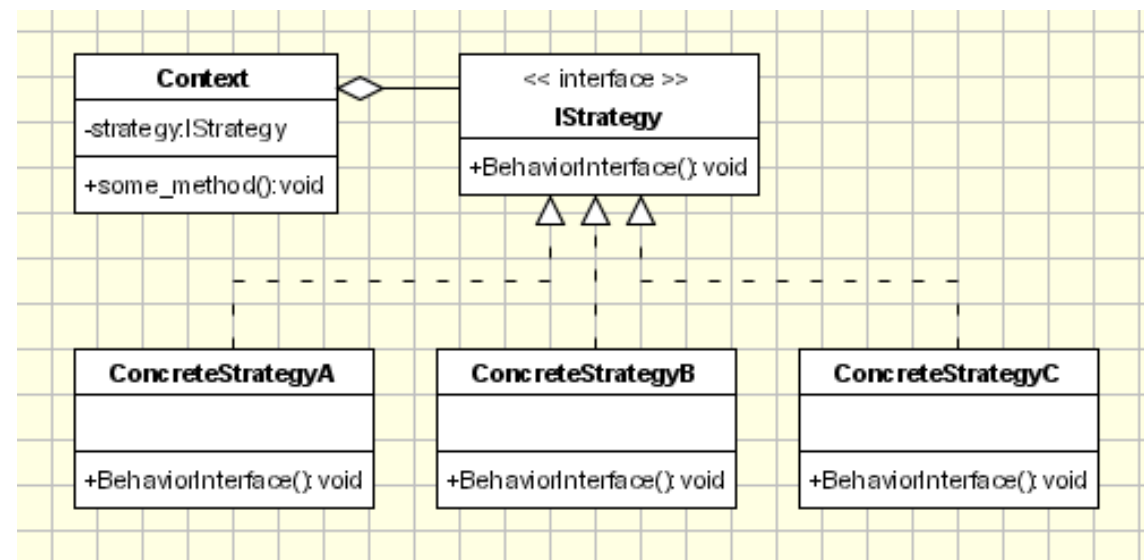
Contexto:

- Uma classe (chamada de contexto) pode se beneficiar de diferentes variantes de um algoritmo.
- Clientes da classe contexto as vezes fornecem versões próprias do algoritmo.

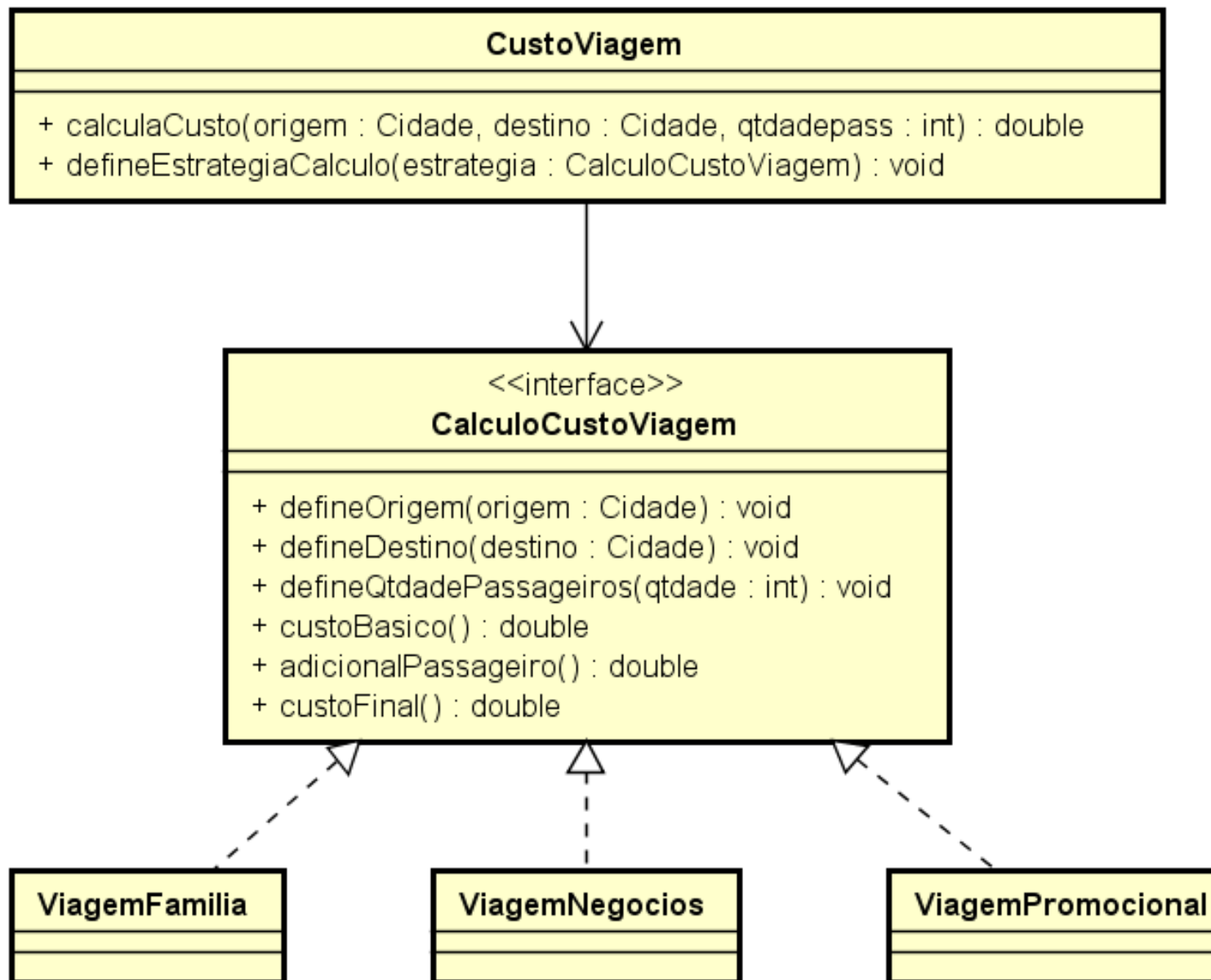
Solução:

- Definir uma interface que seja uma abstração do algoritmo. Chama-se esta interface de Strategy.
- Classes concretas de estratégia implementam a interface Strategy. Cada classe estratégia implementa uma versão do algoritmo.
- O cliente fornece um objeto de estratégia concreto para a classe contexto.
- Sempre que o algoritmo tiver de ser executado, a classe contexto chama os método apropriados do objeto estratégia.

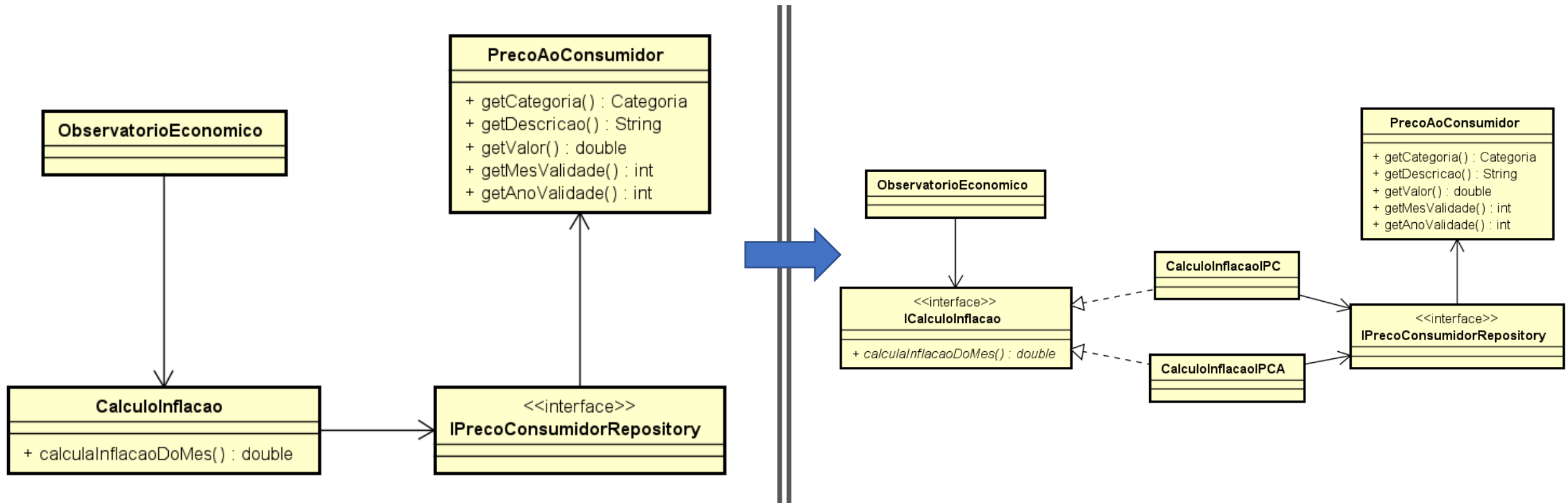
Padrão Strategy: Diagrama



Padrão “Strategy”: exemplo1



Aplicando “Strategy” para obter OCP



Observações

- Observe como usando o Strategy passou a ser possível acrescentar novas estratégias de cálculo de inflação sem necessidade de alterar as demais classes
- Então:
 - “Open”: aberto para novas formas de cálculo (extensões)
 - “Closed”: fechado para modificações nas outras classes



Veja a lista de exercícios

Na sequencia discuta suas conclusões contra a solução apresentada pelo professor

