Padrões de Projeto

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

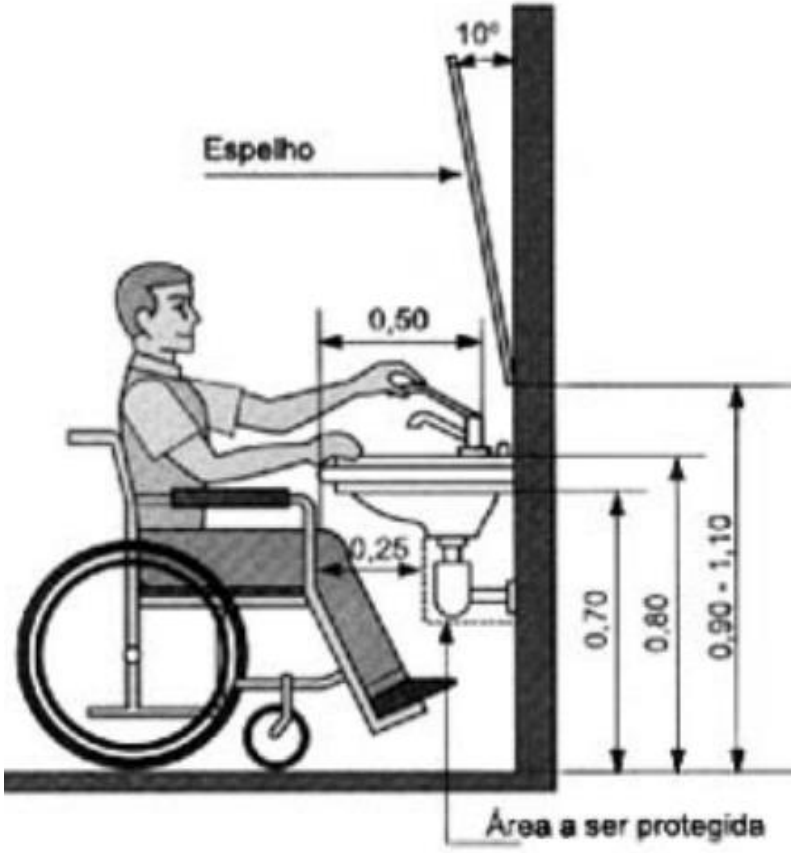
Conteúdo

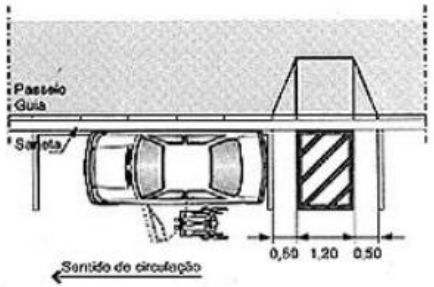
- O que são Padrões de Projeto?
- Para que servem?
- Vantagens/Desvantagens, Pontos Fortes/Fracos
- Exemplos e Alternativas

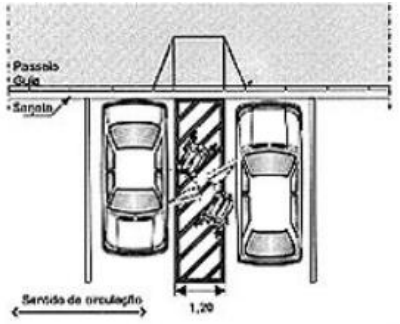
Objetivos

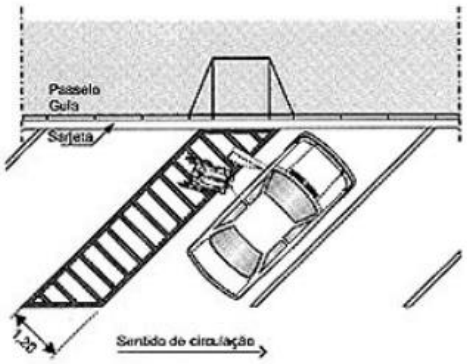
- Conhecer diferentes padrões;
- Entender sua utilidade;
- Pensar no tópico que o grupo pesquisará

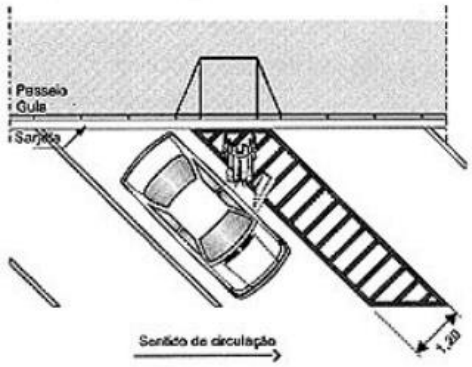
Exemplos

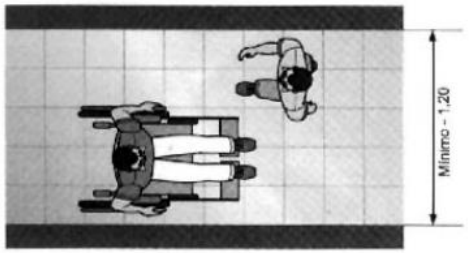

10°
Espelho
0,50
0,25
0,70
0,80
0,90 - 1,10
Área a ser protegida


Passeio
Guia
Sarjeta
0,50 1,20 0,50
Sentido de circulação


Passeio
Guia
Sarjeta
Sentido de circulação
1,20


Passeio
Guia
Sarjeta
1,20
Sentido de circulação


Passeio
Guia
Sarjeta
Sentido de circulação
1,20


Mínimo - 1,20

Exemplos

Motivação

- Projetar software de boa qualidade não é trivial
  - Mistura de específico + genérico
  - Difícil de acertar da primeira vez

- Projetistas experientes usam soluções com as quais já trabalharam no passado

- Padrões de projeto
  - Uso sistemático de:
    - nomes,
    - explicações, exemplos
    - e avaliações

- Durante a última década, uma “comunidade de padrões” foi desenvolvida

O que são Padrões de Projeto?

- É um par problema/solução que pode ser aplicado em novas situações, com recomendações sobre como aplicá-lo e discussão de trade-offs.

- Exemplo:

Nome do Padrão: *Information Expert (Padrão GRASP)*

Problema: Qual é o princípio básico para se atribuir responsabilidades a objetos?

Solução: Atribua responsabilidade à classe que tem a informação necessária

Adicional: consequências/forças

O que são Padrões de Projeto?

- “Cada padrão descreve um problema que ocorre frequentemente em seu ambiente, e então descreve o núcleo de uma solução para o problema, de maneira que você possa usar esta solução um milhão de vezes, sem fazê-la do mesmo jeito duas vezes.”

  Christopher Alexander (Arquiteto e Urbanista), 1977

- “Um padrão é a abstração de uma forma concreta que ocorre muitas vezes em contextos não arbitrários específicos.”

  Riehle and Zullighoven, 1996

- “[É algo que] resolve um problema. É um conceito provado. A solução não é óbvia. Descreve um relacionamento. Os padrões têm um componente humano significativo.”

  Coplien 1996.

O que são Padrões de Projeto?

- É algo comprovado que captura e comunica os conhecimentos das melhores práticas
- Descoberto, não inventado
- Soluções Sucintas e de Fácil Aplicação
  - Capturam, de forma sucinta e facilmente aplicável, soluções de projeto de programas de computador que foram desenvolvidas e evoluíram com o passar do tempo
- Soluções Exaustivamente Refinadas
  - Resultados de um longo processo de projeto, re-projeto, teste e reflexão sobre o que torna um sistema mais flexível, reusável, modular e compreensível
- Soluções Compartilhadas
  - Construídas em grupo
  - Uso de um vocabulário comum

Para que servem?

Design patterns tem vários usos no processo de desenvolvimento de software orientado a objetos:

- Formam um vocabulário comum: isso favorece uma melhor comunicação entre os desenvolvedores, documentação mais completa e melhor exploração das alternativas de projeto.
- Reduzem a complexidade do sistema: define abstrações que estão acima das classes e instâncias;
- Constituem uma base de experiências reutilizáveis: funcionam como peças na construção de projetos de software mais complexos; podem ser considerados como microarquiteturas que contribuem para arquitetura geral do sistema;
- Podem reduzir o tempo de aprendizado de uma determinada biblioteca de classes;
- Quanto mais cedo forem usados, menor tende a ser o retrabalho em etapas mais avançadas do projeto.

Quando usar?

- Em soluções que se repetem com variações.

Não faz sentido o reuso, se o problema só aparece em um determinado contexto.

- Soluções complexas, que requerem vários passos.

Se o número de passos for pequeno, não há a necessidade do uso desses.

- *Design patterns* se aplica bem em situações em que o desenvolvedor está mais interessado na existência de uma solução do que na sua total implementação.

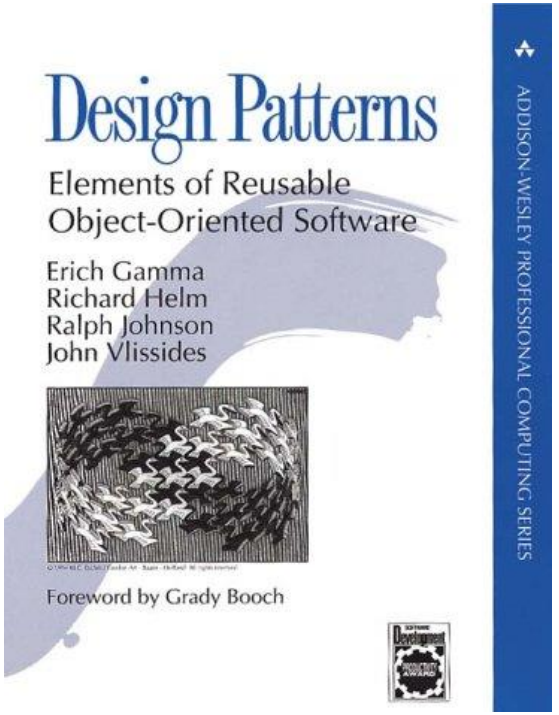
Isto ocorre normalmente quando o domínio do problema é o foco da questão.

Catálogo de soluções

- Um padrão representa o conhecimento de uma pessoa muito experiente em um determinado assunto de uma forma que este conhecimento pode ser transmitido para outras pessoas menos experientes.
- Outras ciências (exemplo: química) e engenharias possuem catálogos de soluções.
- Desde 1995, o desenvolvimento de software passou a ter o seu primeiro catálogo de soluções para projeto de software: o livro GoF.

Gang of Four (GoF)

- E. Gamma and R. Helm and R. Johnson and J. Vlissides. *Design Patterns - Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.



Desafios com o Catálogo de Padrões

- Armazenamento, busca e manutenção da documentação de padrões
- Padrões colecionados em um catálogo precisam ser agrupados
- Projetistas descobrindo novos padrões precisam considerar onde o seu novo padrão se encaixa no catálogo
- Publicação de padrões disponíveis
- Todos os usuários precisam ser atualizados sobre o conteúdo do catálogo

O que são padrões GRASP?

- **G**eneral **R**esponsibility **A**ssignment **S**oftware **P**atterns
- Eles descrevem princípios fundamentais no projeto de objetos, e a atribuição de responsabilidades, expressos como padrões
- A atribuição habilidosa de responsabilidades é extremamente importante;
- Formam as bases de como o sistema será projetado
- A atribuição de responsabilidades ocorre durante a criação de diagramas de interação

5 Padrões GRASP

- 1. Information Expert:

  Qual é o princípio geral para atribuição de responsabilidades a objetos?

- 2. Creator

  Quem deveria ser responsável por criar uma nova instância de alguma classe?

- 3. Controller

  Quem deveria ser responsável por lidar com um evento de entrada do sistema?

- 4. High Cohesion

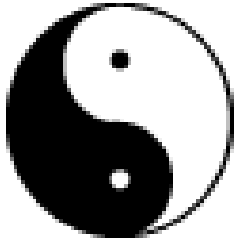
  Como manter a complexidade gerenciável? (não fazer o que não deve)

- 5. Low Coupling

  Como apoiar baixa dependência, baixo impacto em mudança, e aumento de reuso?

Yin e Yang de SE

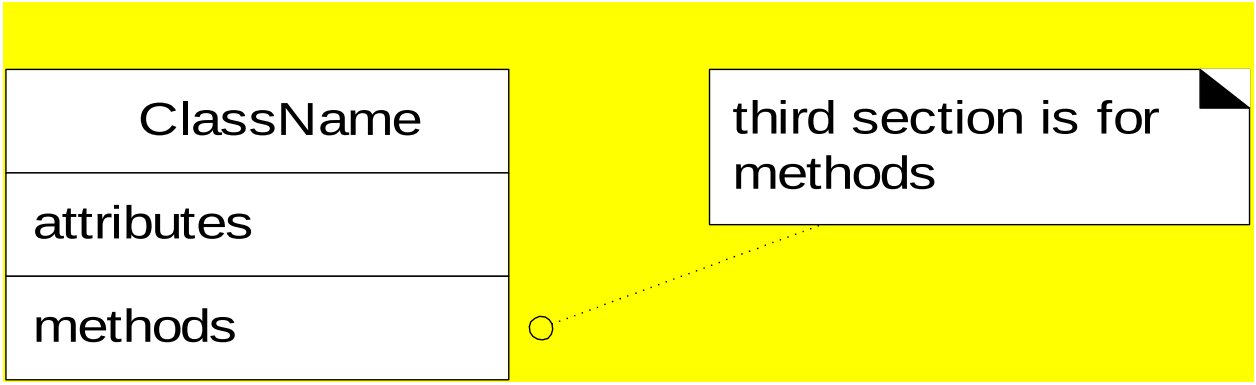
Coesão ruim normalmente leva à **Acoplamento** ruim, e vice-versa!



Cohesion and coupling *are the yin and yang of software engineering*

Faça o que deve ser feito, e do modo mais independente possível…

A Notação UML para Diagrama de Classes



Pattern: Information Expert

Problema que resolve: Qual é o princípio geral para atribuição de responsabilidades a objetos?

Solução: Atribua uma responsabilidade à classe que tem a informação necessária

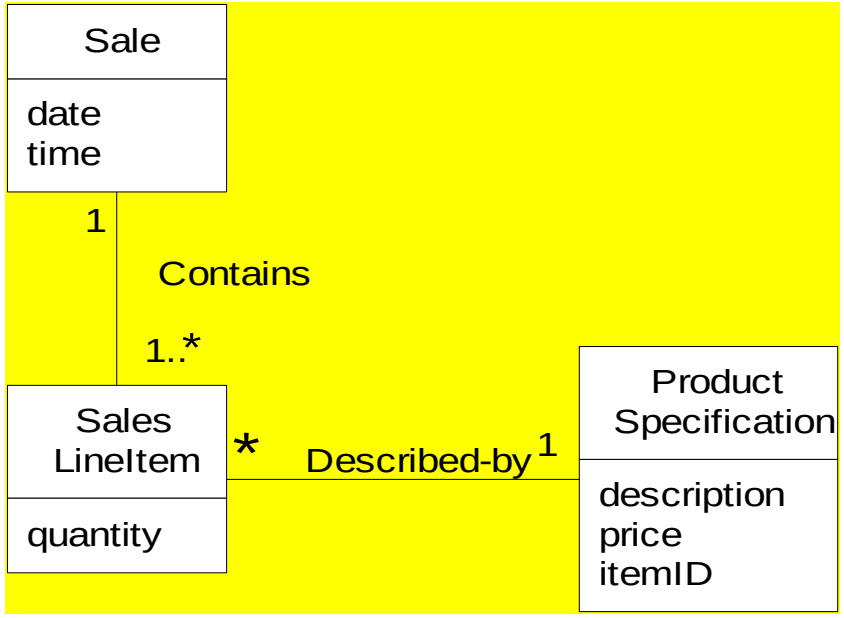
Exemplo:

Pattern: Information Expert

Problema que resolve: Qual é o princípio geral para atribuição de responsabilidades a objetos?

Solução: Atribua uma responsabilidade à classe que tem a informação necessária

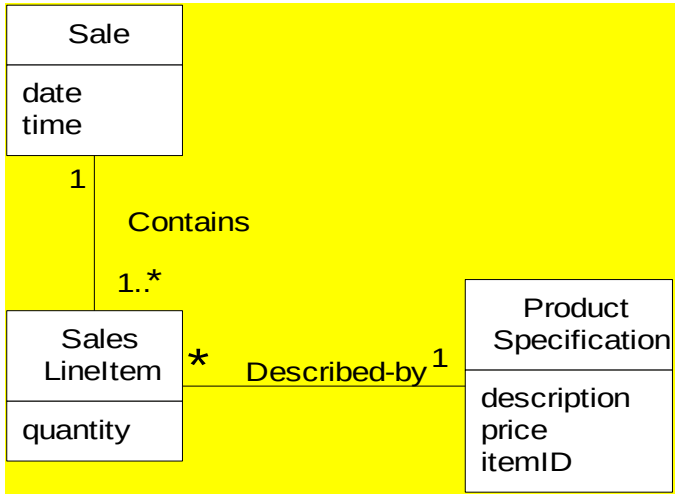
Exemplo: Em uma aplicação de vendas, algumas classes necessitam conhecer o total geral de *Sale*



Quem deveria ser responsável por conhecer o total geral de sale?

Que informação é necessária para determinar o total geral?

- É necessário conhecer todas as instâncias de *SalesLineItem* de uma *Sale* e a soma de seus sub-totais.
- Uma instância de *Sale* contém esses elementos.
- *Sale* é uma classe de objetos adequada para esta responsabilidade.
- Ela é uma **Information Expert** para esse trabalho

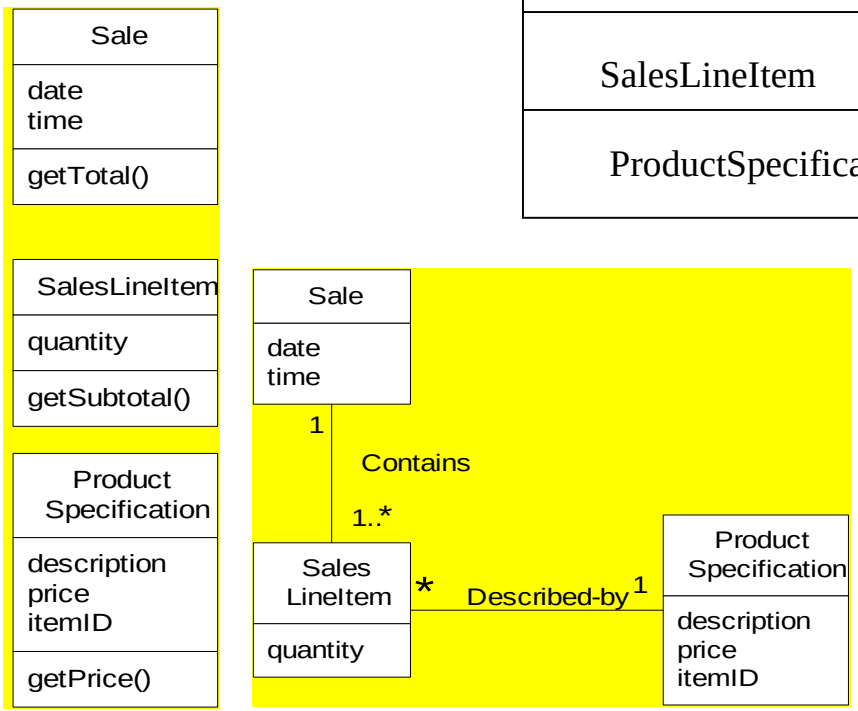


Diagramas parciais de Interação e de Classe

Responsabilidades…

- Para responder à responsabilidade de conhecer e responder **o total de *sale***, 3 responsabilidades devem ser atribuídas a 3 classes de design de objetos:

Design Class	Responsibility
Sale	Knows sale total
SalesLineItem	Knows line item sub total
ProductSpecification	Knows product price



Padrão: Information Expert

- ***Information Expert*** é um princípio básico usado para guiar continuamente o design orientado a objetos
- A resposta a uma responsabilidade geralmente requer informação que está espalhada em diferentes classes de objetos
  - Isso significa que há muitos information experts “parciais” que colaborarão
  - Eles precisarão interagir via mensages para compartilhar o trabalho

Questões Relativas ao Uso de Padrões

- Existe algum padrão que trata um problema similar?
- O contexto do padrão é consistente com o problema real?
- As consequências do uso de padrão são aceitáveis?
- O padrão tem uma solução alternativa que pode ser mais aceitável?
- Existe uma solução mais simples?
- Existem algumas restrições que sejam impostas pelo ambiente de software que está sendo usado?

Perigos do Uso de Padrões

- O uso de padrões de uma maneira descontrolada pode originar projetos sobrecarregados.

Desenvolvedores:

- Precisam de tempo para entender os catálogos de padrões relevantes
  - Precisam de fácil acesso a catálogos relevantes
  - Precisam ser treinados no uso de padrões

Referências

- Larman, C. (2002) *Applying UML and Patterns – An Introduction to Object Oriented Analysis and Design and the Unified Process*, Prentice-Hall Inc.
- Muller, P.A. (1997) *Instant UML*,Wrox Press Ltd.

Atividade

- Investigar o tópico escolhido pelo Grupo (ou trabalhar na implementação)