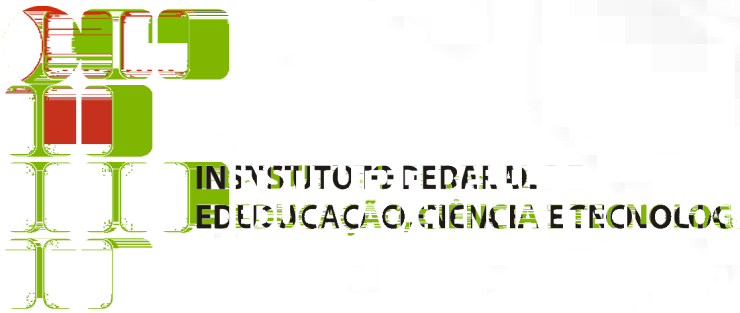
INF011 – Padrões de Projeto

01 - Introdução

"andro "antos Andrade
sandroandrade@ifba.edu.br

Instituto Federal de $ducação% &i'ncia e Tecnologia da (a)ia
e*artamento de Tecnologia $letro+$letr,nica
Graduação Tecnológica em Análise e esen!ol!imento de "istemas



Introdução

- Projetar *software* orientado a objetos é difícil
- Projetar *software* reutilizável é mais difícil ainda
- O projeto deve ser específico ao problema, porém genérico o suficiente para acomodar futuras mudanças
- É difícil obter um projeto reutilizável e flexível na primeira tentativa
- Projetistas novatos levam um tempo para entender o que é um bom projeto orientado a objetos

Introdução

- O que os projetistas experientes fazem:
  - Reutilizam soluções que funcionaram no passado
  - Muitos sistemas orientados a objetos compartilham padrões de funcionamento das classes e da comunicação entre objetos
  - Estes padrões tornam o projeto mais flexível, elegante e reutilizável
  - O projetista aplica o padrão de projeto sem ter que o re-descobrir
  - Novelistas e dramaturgos aplicam padrões constantemente
  - O mesmo vale para *software*

Introdução

- Exemplos:
  - “Represente o estado como um objeto”
  - “Decore os objetos de modo que funcionalidades possam ser facilmente adicionadas ou removidas”
- Se você conhece o padrão uma série de decisões de projeto surgem automaticamente
- Um padrão de projeto registra uma determinada experiência bem sucedida em projeto de *software*
- Cada padrão sistematicamente nomeia, explica e avalia um projeto importante e recorrente

Introdução

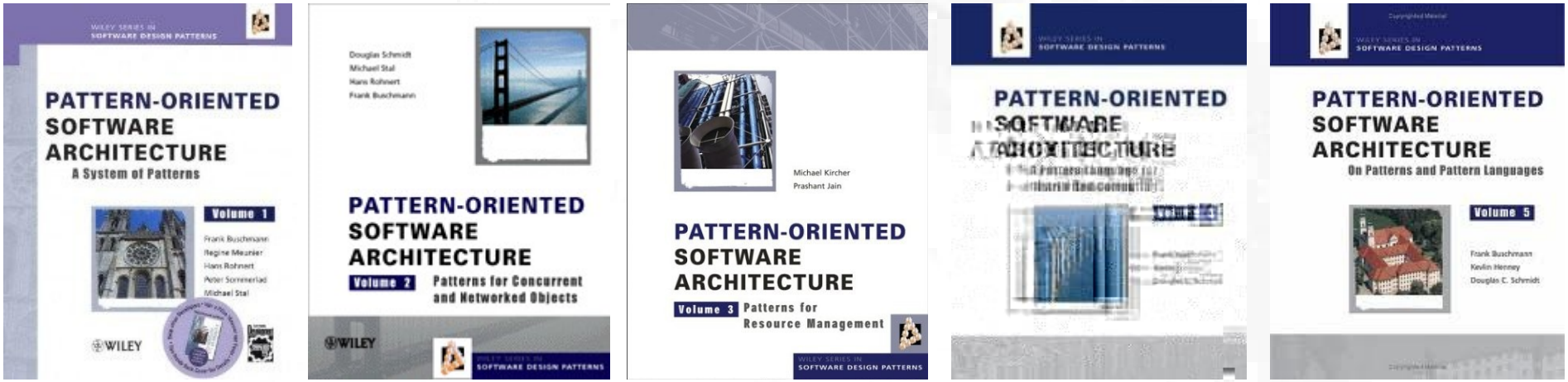
- Os padrões de projeto facilitam a definição de um projeto “correto” em um tempo reduzido
- *Christopher Alexander*: “cada padrão descreve um problema que ocorre frequentemente em nosso ambiente e descreve o núcleo da solução para o problema, de uma forma que ela possa ser utilizada inúmeras vezes”

Introdução

- Um padrão possui quatro elementos:
  - Nome: identificador utilizado para descrever, com uma ou duas palavras, o problema, sua solução e consequências
  - Pro-lema: descreve quando aplicar o padrão e o contexto do problema
  - "olução: descreve os elementos que compõem o projeto, seus relacionamentos, responsabilidades e colaborações. Não descreve uma implementação ou projeto concreto em particular
  - &onse.u'ncias: resultados e *trade-offs* da aplicação do padrão

Introdução

- Os padrões apresentados são descrições de classes e objetos inter-relacionados que solucionam um problema de projeto em um contexto particular
- Outros padrões estão disponíveis para solucionar problemas de concorrência, computação distribuída, tempo-real e aspectos específicos de domínio



/ &atálogo de Padrões

- *Abstract Factory*
- *Adapter*
- *Bridge*
- *Builder*
- *Chain of Responsibility*
- *Command*
- *Composite*
- *Decorator*
- *Facade*

/ &atálogo de Padrões

- *Factory Method*
- *Flyweight*
- *Interpreter*
- *Iterator*
- *Mediator*
- *Memento*
- *Observer*
- *Prototype*
- *Proxy*

/ &atálogo de Padrões

- *Singleton*
- *State*
- *Strategy*
- *Template Method*
- *Visitor*

/ &atálogo de Padrões

		Purpose		
		Creational	**Structural**	**Behavioral**
Scope	**Class**	Factory Method (107)	Adapter (139)	Interpreter (243) Template Method (325)
	Object	Abstract Factory (87) Builder (97) Prototype (117) Singleton (127)	Adapter (139) Bridge (151) Composite (163) Decorator (175) Facade (185) Proxy (207)	Chain of Responsibility (223) Command (233) Iterator (257) Mediator (273) Memento (283) Flyweight (195) Observer (293) State (305) Strategy (315) Visitor (331)

/ &atálogo de Padrões

- Alguns padrões são utilizados em conjunto. Ex: *Composite* com *Iterator* ou *Visitor*
- Alguns outros são alternativas. Ex: *Prototype* ou *Abstract Factory*
- Alguns padrões apresentam projeto similar, embora tenham diferentes propósitos. Ex: *Composite* e *Decorator*

/ &atálogo de Padrões

- Como os padrões resolvem os problemas ?
  - Encontrando objetos apropriados: a decomposição é influenciada por fatores tais como encapsulamento, granularidade, dependências, flexibilidade, desempenho, evolução, reutilização, etc. Frequentemente surgem classes que não possuem correspondentes no negócio
  - Determinando a granularidade do objeto
  - Especificando as interfaces dos objetos: um tipo denota uma interface em particular. Um objeto pode ter muitos tipos e diferentes objetos podem ter um mesmo tipo. Ligação dinâmica e polimorfismo

/ &atálogo de Padrões

- Como os padrões resolvem os problemas ?
  - Especificando a implementação de objetos: notação para classe. Instanciação (seta tracejada). Herança de classe (implementação). Classes abstratas (em itálico). Classes *mix-in*. Diferença entre a *classe* (implementação) e o *tipo* (interface) do objeto

INF011 – Padrões de Projeto

01 - Introdução

"andro "antos Andrade
sandroandrade@ifba.edu.br

Instituto Federal de $ducação% &i'ncia e Tecnologia da (a)ia
e*artamento de Tecnologia $letro+$letr,nica
Graduação Tecnológica em Análise e esen!ol!imento de "istemas