



Frameworks

José Osvano da Silva, PMP



Sumário

- Introdução
- Conceito
- Exemplo
- Frameworks de Aplicação
- Benefícios
- Tipos de Framework
- Vantagens
- Desvantagens
- Exemplos

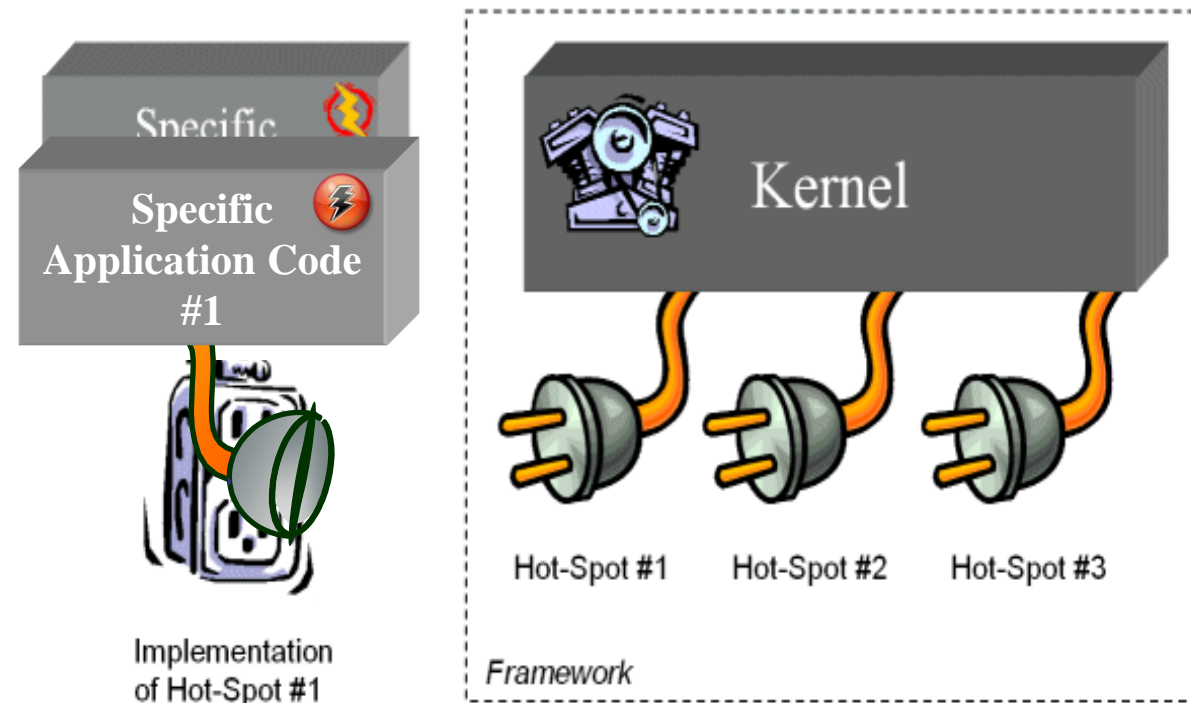


Introdução

- Um Framework é uma aplicação “**semi-completa**”, **reutilizável**, que pode ser **especializada** para produzir **aplicações** personalizadas

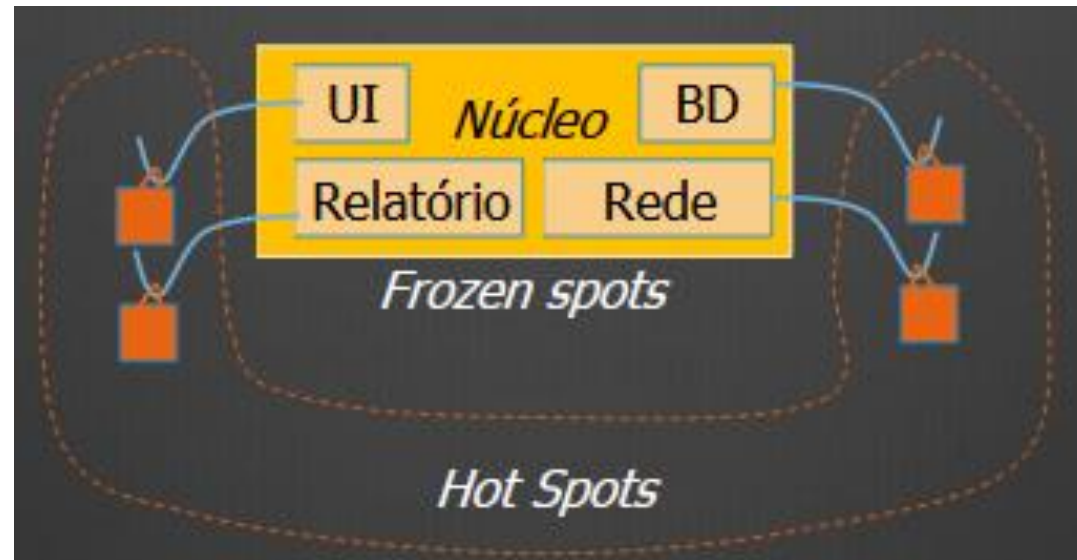
Conceito

- Um framework separa o que é fixo (*frozen-spots*) do que é variável (hot-spots)



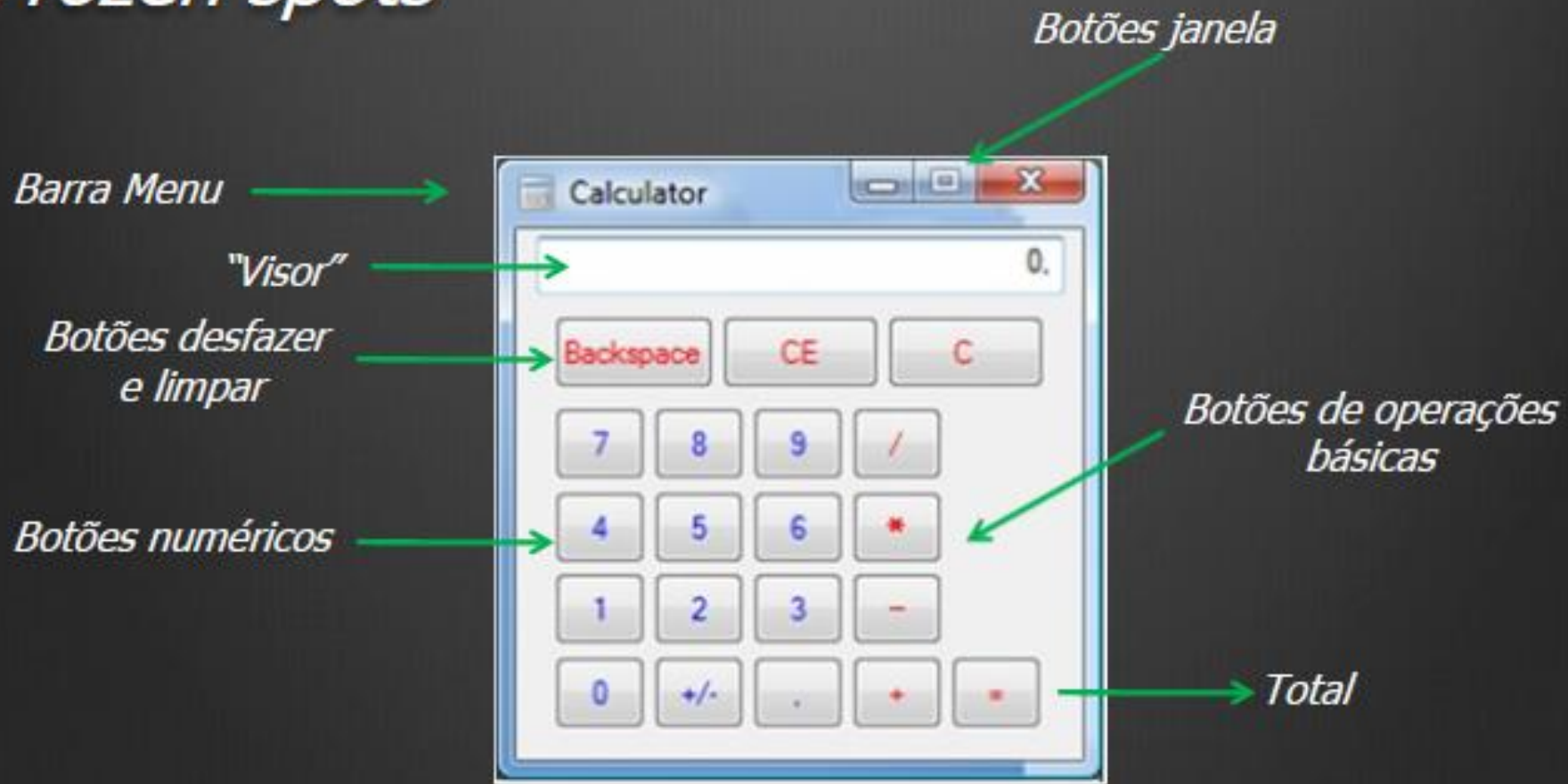
Conceito

- Identificam-se similaridades entre aplicações e cria-se um núcleo invariante com ganchos para extensão

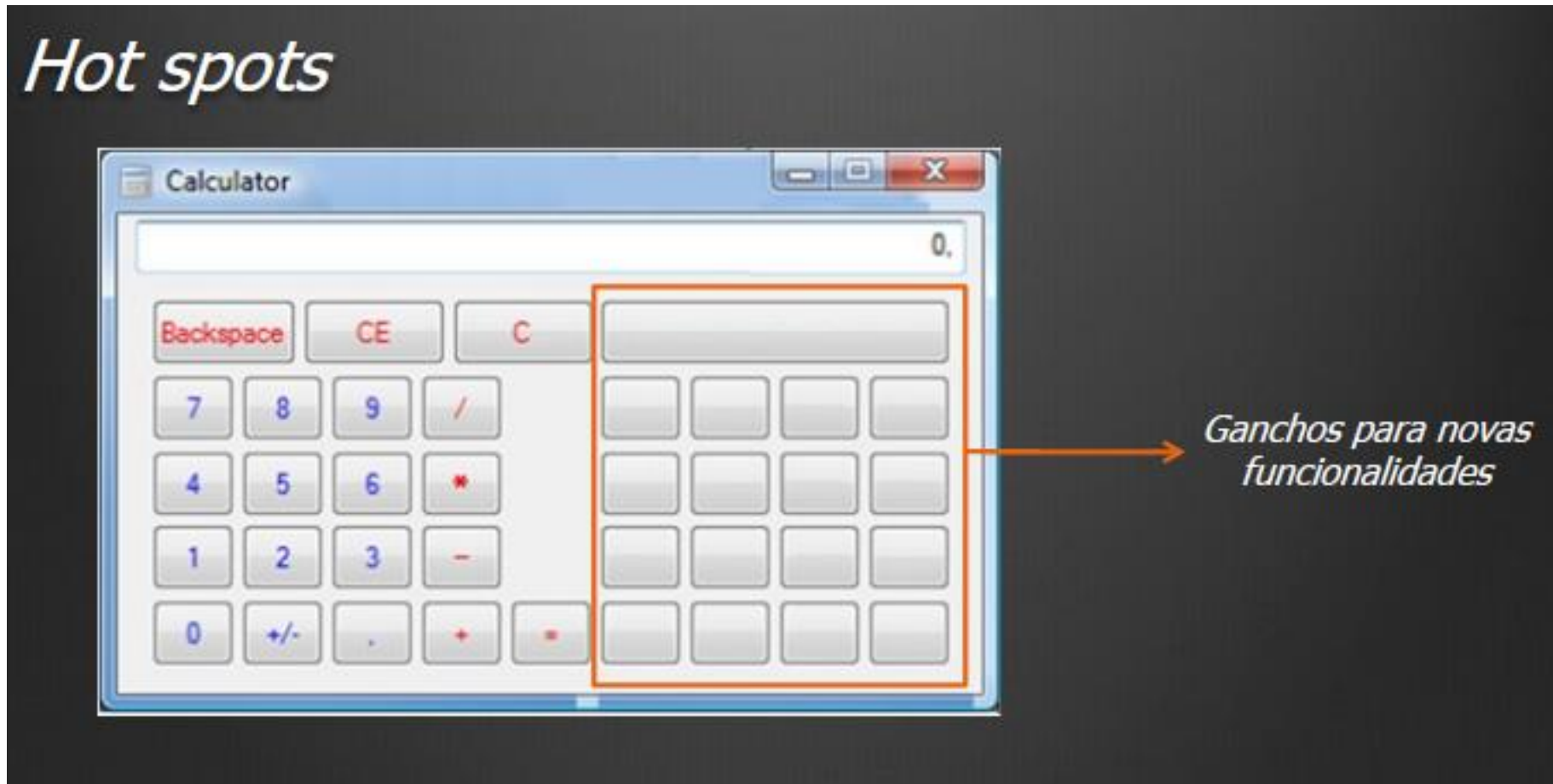


Exemplo

Frozen spots



Exemplo



Frameworks de Aplicação

- Framework de aplicação

- Projeto constituído de uma coleção de classes concretas e abstratas, e de interfaces entre elas

- Instância do framework

- Implementada pela adição de detalhes específicos e pela instanciação das classes abstratas

- Frameworks

- Entidades “relativamente” grandes que podem ser reutilizadas



Benefícios

- Os principais benefícios dos Frameworks de Aplicação são:
 - Modularidade
 - Reusabilidade
 - Extensibilidade
 - Inversão de controle



Benefícios: Modularidade

- Encapsulam comportamentos “voláteis” atrás de interfaces bem definidas;
- Potencializam a qualidade de software através da localização explícita de mudanças nos aplicativos;

Benefícios: Reusabilidade

- Interfaces estáveis permitem definir componentes genéricos que podem ser reutilizados em diversas aplicações;
- Com maturidade dos componentes, tem-se:
 - Qualidade
 - Desempenho
 - Confiança no funcionamento.

Benefícios: Extensibilidade

- Métodos gancho permitem estender o próprio framework
 - Adaptações para construir novos tipos de aplicações
- Os ganchos permitem personalizar as aplicações para os propósitos específicos, sem alterar a estabilidade do núcleo do framework.



Benefícios: Inversão de Controle

- A arquitetura “executável” do framework é caracterizada pela inversão de controle
- Isto permite que interfaces definam comportamentos específicos ativados como objetos de manipulação de eventos, que são invocados pelo mecanismo de disparo de eventos e ações do Framework
 - Princípio de Hollywood

Tipos de Framework: Perspectiva de Reuso

■ White Box (Caixa Branca)

□ Baseado principalmente em Herança e Ligação Dinâmica

- Realizando Herança de classes fundamentais
- Sobrescrevendo métodos gancho pré-definidos
- Requer conhecimento do código do Framework;
- Permitindo assim entendê-lo sem prejuízo ao núcleo.

Tipos de Framework: Perspectiva de Reuso

■ Black Box (Caixa Preta)

- Baseado na definição de interfaces para componentes que podem ser “públicos” ao framework via composição de objetos
 - Componentes que respeitam determinada interface (contrato)
 - Componentes integrados ao Framework
- Requer apenas conhecimento da interface de composição do Framework
 - Pois o núcleo mantém-se inalterado.

Tipos de Framework: Perspectiva de Reuso

■ Grey Box (Caixa Cinza)

- Mescla reuso caixa branca com reuso caixa preta.
 - Em geral é mais desejável:
 - Caixa Branca para extensão do Framework;
 - Caixa Preta para a criação de aplicações.

Vantagens

- Com o framework pronto, benefícios

- Redução de custos
 - Redução de time-to-market

- Motivos

- Maximização de re-uso (análise, design, código, testes)
 - Reutilização de design feito por outros pode transferir conhecimento e experiência para o usuário do framework
 - Desenvolvedores adicionam valor em vez de reinventar a roda
 - Menos manutenção
 - Fatoração de aspectos comuns a várias aplicações
 - Uso de herança permite corrigir todas as aplicações com a troca de uma classe-mãe
 - Cuidado com o "Fragile Base Class Problem" onde troca da classe-mãe quebra as filhas
 - Melhora do código (menos defeitos) devido ao uso em várias aplicações

Vantagens

■ Vantagens

- Diminuição de linhas de código na aplicação
- Melhor consistência e compatibilidade entre aplicações
- Conhecimento sobre o domínio da aplicação é mantido dentro da organização
- Alavancagem do conhecimento de especialistas
 - Framework oferece uma forma de empacotar o conhecimento de especialistas sobre domínios de problemas
 - Assim, não se perde o conhecimento com a saída de especialistas e o conhecimento pode ser usado/estudado sem a presença do especialista
 - Resultado: criação de patrimônio estratégico da empresa (Strategic Asset Building)

Desvantagens

- Construir um framework é complexo
 - Re-uso não vem sozinho: deve ser planejado
 - É mais complexo e demora mais fazer uma aplicação tendo que construir um framework em vez de fazer a aplicação do zero
- Documentação é essencial para o usuário (desenvolvedor) poder utilizar o framework
- Dificuldade para manter compatibilidade com versões anteriores
 - Frameworks se tornam mais maduros com o passar do tempo e as aplicações devem evoluir em paralelo
- Flexibilidade e generalização do framework podem trabalhar contra sua eficiência em algumas aplicações

Desvantagens

- Benefícios são realizados em longo prazo
 - Quem pode pensar em longo prazo quando se está competindo "On Internet time"?
 - Poucas empresas
 - Uma empresa aeroespacial demorou anos para fazer frameworks e começou a ter retorno na quarta missão
- Precisa-se modificar o processo de desenvolvimento e criar novos incentivos.

Exemplos: Container IoC

- Spring – <http://www.springsource.org/>
- Guice – <http://picocontainer.codehaus.org/>
- PicoContainer – <http://code.google.com/p/google-guice/>
- EJB – <http://www.oracle.com/technetwork/java/javaee/ejb/index.html>
- CDI – <http://www.oracle.com/technetwork/articles/java/cdi-javaee-bien-225152.html>

Exemplos: Persistência de Dados

- Hibernate – <http://www.hibernate.org/>
- Spring Data – <http://www.springsource.org/>
- iBatis – <http://www.mybatis.org/>
- Torque – <http://db.apache.org/torque/>
- Castor – <http://www.castor.org/>
- Cayenne – <http://cayenne.apache.org/>

Exemplos: Gerenciador de transação

- Spring AOP transaction – <http://www.springsource.org/>
- EJB (JTA) – <http://www.oracle.com/technetwork/java/javaee/ejb/index.html>

Exemplos: Agendamento de tarefas

- Timer Task JSE –
<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/threads/timer.html>
- Quarts – <http://quartz-scheduler.org/>
- EJB Timer Service –
<http://docs.oracle.com/javaee/6/tutorial/doc/bnboy.html>

Exemplos: Validação

- Commons Validator – <http://commons.apache.org/validator/>
- Java Bean Validator – <http://docs.oracle.com/javaee/6/tutorial/doc/gircz.html>

Exemplos: Chamadas assíncronas

- JSE Executor Services –
<http://docs.oracle.com/javase/tutorial/essential/concurrency/exinter.html>
- EJB Asynchronous Method Invocation –
<http://docs.oracle.com/javaee/6/tutorial/doc/gkkqg.html>
- EJB Message-Driven Bean –
<http://docs.oracle.com/javaee/6/tutorial/doc/gipko.html>

Exemplos: Kit de interface gráficas desktop

- Swing – <http://docs.oracle.com/javase/tutorial/uiswing/>
- SwingX – <http://swingx.java.net/>
- JGoodies – <http://www.jgoodies.com/>
- SWT – <http://www.eclipse.org/swt/>
- SWT OPAL – <http://code.google.com/a/eclipselabs.org/p/opal/>

Exemplos: RAD interface gráficas desktop

- Eclipse – <https://eclipse.org/>
- NetBeans – <https://netbeans.org/>

Exemplos: Framework web

- Click – <http://click.apache.org/>
- Wicket – <http://wicket.apache.org/>
- Spring MVC – <http://www.springsource.org/>
- Struts – <http://struts.apache.org/>
- Mentawai – <http://www.mentaframework.org/>
- JSF – <http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>
- VRaptor – <http://vraptor.caelum.com.br/>
- Angular JS - <https://angularjs.org/>

Exemplos: Framework web RIA

- GWT – <https://developers.google.com/web-toolkit/>
- Vaadin – <https://vaadin.com/home>
- ZK Framework – <http://www.zkoss.org/>

Exemplos: Provedores de JSF

- Mojarra – <http://javaserverfaces.java.net/>
- MyFaces – <http://myfaces.apache.org/>
- ADF – <http://www.oracle.com/technetwork/developer-tools/adf/overview/index.html>

Exemplos: Componentes JSF

- RichFaces – <http://www.jboss.org/richfaces>
- PrimeFaces – <http://primefaces.org/>
- OpenFaces – <http://openfaces.org/>
- IceFaces – <http://www.icesoft.org/>
- EasyFaces – <http://www.easyfaces.com.br>
- Gmaps4jsf – <http://code.google.com/p/gmaps4jsf/>

Exemplos: JEE web container standalone

- Tomcat – <http://tomcat.apache.org/>
- Jetty – <http://jetty.codehaus.org/jetty/>
- JBoss – <http://www.jboss.org/>
- Wildfly - <http://www.wildfly.org/>

Exemplos: Relatórios

- iReport – <http://jasperforge.org/projects/ireport>
- DynamicReport – <http://dynamicreports.sourceforge.net>
- Birt – <http://www.eclipse.org/birt/phoenix/>
- iText – <http://itextpdf.com/>
- PDFBox – <http://pdfbox.apache.org/>

Exemplos: Gerador de gráficos charts

- JFreechart – <http://www.jfree.org/jfreechart/>
- HighCharts – <http://www.highcharts.com/>

Referências

- SOMMERVILLE, Ian, **Engenharia de Software**, 8ª Edição, São Paulo, Editora Pearson Prentice Hall, 2007.
- PRESSMAN, R. S. **Engenharia de software**: uma abordagem profissional. 7ª Edição. Porto Alegre: AMGH, 2011. 780 p.
- GAMMA, Erich. **Padrões de projeto: soluções reutilizáveis de software orientado a objeto** / Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides; trad. Luiz A. Meirelles Salgado. - Porto Alegre: Bookman, 2000.

Dúvidas



José Osvano da Silva
joseosvano@unipac.br