

Programação Orientada a Objetos

Reúso em POO

André Santanchè
Instituto de Computação - UNICAMP
Abril 2015

Reúso de Software

- “Software reuse is the process of creating software systems from existing software rather than building software systems from scratch”. (Krueger, 1992)

Por que reusar?

- Aumento de produtividade
- Redução de custos de desenvolvimento e manutenção
- Reuso pode promover sucessivas melhorias no produto
- Frequência de reuso pode indicar qualidade

Reúso Sistematizado

- Aspectos técnicos:
 - definição de estratégias para reusar
 - aplicação de padrões na produção
 - documentação e indexação dos artefatos
 - mecanismo de adaptação do conteúdo

Aspectos de Reuso

(Mili, 1995)

- Reúso de software envolve:
 - desenvolvimento de software cujo projeto é reusável
 - desenvolvimento com software reusável
- Reúso de software inclui:
 - reúso de produtos desenvolvidos
 - reúso do processo para o desenvolvimento
- Abordagens de reuso variam:
 - blocos de construção (reuso do produto)
 - *genative* ou *reusable processor* (reuso de processo)

Taxonomia de Reúso de Software

(Krueger, 1992)

- Linguagens de Alto Nível
- Garimpagem (*Scavenging*) de Projeto e Código
- Componentes de Software
- Esquemas de Software (ou Padrões de Programação Reusáveis)
- Geradores de Aplicação
- Linguagens de Altíssimo Nível (*Very High-Level Languages* - VHLL)
- Sistemas de Transformação
- Arquiteturas de Software

Reúso Sistematizado em OO

- Reúso de código
 - Classes em bibliotecas e frameworks
 - Herança
 - Agregação
- Reúso de projeto (design)
 - Princípios de projeto
 - Padrões de projeto (design patterns)

Princípios de Projeto

- Projetos para sistemas confiáveis e extensíveis, exigem atenção para diversos aspectos relacionados à sua construção
- Princípios para a elaboração de projetos
 - criados sobre a tecnologia de orientação a objetos
 - baseados em diversos casos de sucesso
 - aumentam as chances de se produzir projetos de qualidade.

Referências

- AG Communication Systems. **Examples to Accompany: Design Patterns Elements of Reusable Object-Oriented Software**, 1999.
- Abowd, G. D., Allen, R., Garlan, D. **Formalizing style to understand descriptions of software architecture**. ACM Trans. Softw. Eng. Methodol., ACM Press, 1995, 4, 319-364.
- Alexander, Christopher; Ishikawa, Sara; Silverstein, Murray. **A Pattern Language: Towns, Buildings, Construction**. Oxford University Press, 1977.
- Krueger, C. W. **Software Reuse**. ACM Comput. Surv., ACM Press, 1992, 24, 131-183.
- Mcilroy, M. D. Naur, P. & Randell, B. (ed.) **Mass Produced Software Components**. Software Engineering: Report of a conference sponsored by the NATO Science Committee, 1968.

Referências

- Mili, H.; Mili, F. & Mili, A. **Reusing Software: Issues and Research Directions**. IEEE Transactions on Software Engineering, 1995, 21, 528-562.
- Shaw, M. **Abstraction Techniques in Modern Programming Languages**. IEEE Software, 1984, 1, 4, 10-26.
- Sommerville, I. (2007) **Software Engineering**, 8th. ed. Addison Wesley.

André Santanchè

<http://www.ic.unicamp.br/~santanche>

License

- These slides are shared under a Creative Commons License. Under the following conditions: Attribution, Noncommercial and Share Alike.
- See further details about this Creative Commons license at: <http://creativecommons.org/licenses/by-nc-sa/3.0/>