

Programação Orientada por Objetos

Objetivos e Filosofia



Roberto da Silva Bigonha
Mariza A. S. Bigonha

Agosto de 2019

Todos os direitos reservados
Proibida a cópia sem autorização dos autores



Qualidade do Software

Um dos principais objetivos da Engenharia de Software é ajudar a produção de **software de boa qualidade**.

Fatores de Qualidade do Software:

Externos

Internos



...Qualidade do Software

☐ Fatores de Qualidade Externos

- Correção
- Robustez
- Extensibilidade
- Reusabilidade
- Eficiência
- Compatibilidade
- Facilidade de uso
- Portabilidade
- Integridade
- Verificabilidade



...Qualidade do Software

☐ Fatores de Qualidade Internos:

- Modularidade
- Legibilidade
- Manutenibilidade



Fatores Externos de Qualidade

Correção:

- ☐ Correção é a propriedade de um produto de software executar exatamente sua função, conforme definida pelos requisitos e especificação.
- ☐ Correção garante o funcionamento no que foi previsto.

Robustez

- ☐ Robustez é a propriedade de um software funcionar mesmo em condições anormais.
- ☐ Robustez garante funcionamento no que **não** foi previsto.
- ☐ Capacidade de *degradação suave*.



...Fatores Externos de Qualidade

Eficiência

- ☐ Eficiência é o bom uso dos recursos de hardware, tais como processador, memória, dispositivos de comunicação.
- ☐ Correção X Eficiência

Extensibilidade

- ☐ Extensibilidade é uma medida da facilidade com que o software pode ser adaptado para atender a mudanças na sua especificação.
- ☐ Extensibilidade é propriedade essencial em *Programação em Ponto Grande*.
- ☐ Princípios que facilitam extensibilidade:
 - Simplicidade do projeto
 - Descentralização: módulos autônomos



...Fatores Externos de Qualidade

Reusabilidade

- ☐ Reusabilidade é a propriedade de um software ser **usável** em novas aplicações.

Compatibilidade

- ☐ Compatibilidade é uma medida da facilidade com que um software pode ser combinado com outros.
- ☐ Compatibilidade implica em projeto homogêneo e padronização, por exemplo:
 - arquivo padrão no UNIX
 - estrutura de dados única do LISP
 - interface padronizada do Smalltalk



...Fatores Externos de Qualidade

Facilidade de Uso

- ☐ Facilidade de uso inclui preparação de dados de entrada, interpretação de resultados, recuperação de erros de uso, etc.

Portabilidade

- ☐ Portabilidade é uma medida da facilidade de transporte de um software para vários hardware ou ambientes de programação.

Integridade

- ☐ Integridade é a capacidade de um software de proteger seus componentes contra acesso ou modificação não autorizada.

Verificabilidade

- ☐ É uma medida da facilidade de se preparar procedimentos de aceitação, dados para testes, etc, durante a fase de validação ou de operação.



Custo de Manutenção de Software

- ☐ Manutenção é modificação para atender mudanças no mundo externo
- ☐ Manutenção é remoção de erros, que não deviam estar lá.
- ☐ Cerca de 70% do custo de um software é manutenção.
- ☐ Distribuição do custo de manutenção apurada em 487 instalações:
 - mudanças na especificação: 41.8%
 - mudanças no formato dos dados: 17.4%
 - consertos de emergências: 12.4%
 - depuração: 9.0%
 - mudanças no hardware: 6.2%
 - atualização da documentação: 5.5%
 - melhoria na eficiência: 4%
 - outras: 3.4%
- ☐ Lintz, B.P. & Swanson, E.B., *Software Maintenance: A User/Management Tug of War*, Data Management, pp. 26-30, april 1979.



Mudança de Paradigma



Métodos Tradicionais de Projeto

- ☐ Projeto tem por objetivo construir a ponte sobre o **gap semântico**
- ☐ Projeto Estruturado Top-Down:
Problema é função única, que é decomposta em componentes diretamente implementáveis.
- ☐ Projeto Orientado por Dados:
Projeto baseado no mapeamento dos dados de entrada para os de saída.



Desenvolvimento TOP-DOWN

- ☐ Desenvolvimento **top-down** gera arquitetura baseada somente na **função** do sistema:
 - Especificação define a **função**.
 - Processo **top-down** começa com um *comando* que expressa a **função** principal.
 - Cada passo refina ou decompõe **funções**.
 - Decomposição pára quando o nível de abstração da **função** refinada for diretamente implementável.
- ☐ Portanto:
 - **Top-Down** é a decomposição da **função** do sistema como uma árvore de subfunções.
 - Cada função da árvore é implementada como uma **rotina**.
 - **Top-Down** é baseado somente na **abstração** de comandos e expressões.



Pontos Positivos do Top-Down

- Disciplina o pensamento de forma lógica e organizada.
- Permite o desenvolvimento de forma ordenada.
- Oferece uma forma sistemática para quebrar a aparente complexidade inicial.



Pontos Negativos do Top-Down

- ☐ Método não leva em consideração a natureza evolutiva dos sistemas de software.
- ☐ A idéia de um sistema ser caracterizado por uma única função é questionável.
- ☐ O uso da função como base negligencia o aspecto *Estrutura de Dados*.
 - Método não favorece reúso de software.
 - Dificuldades com Continuidade.



Top-Down X Interface Externa

- ☐ A função não é a parte mais **estável** de um sistema.
A interface do sistema é apenas um componente.
- ☐ Programa com duas versões: **batch** e **interativa**.
 - **Versão batch:**
Leia valores de entrada;
Compute os resultados;
Imprima resultados;
 - **Versão interativa:**
if novos dados fornecidos then Leia dados; Armazene dados;
elsif há pedidos sobre velhos dados then
 Recupere informação; Imprima resultado
elsif resultados pedidos then
 if informação está disponível then
 Recupere resultados pedidos; Imprima resultados;



Top-Down X Decisões de Projeto

- ☐ **Top-Down** fixa prematuramente a interface externa, porque responde sempre à pergunta

O que o sistema faz por mim?
em vez de
Sobre o que o sistema faz o que?
- ☐ A escolha da estrutura deveria ser baseada em critérios como *espaço X tempo*.



...Top-Down X Decisões de Projeto

❑ **Top-Down** dá muita ênfase às relações temporais:

▪ **Opção 1 para um Compilador:**

Leia fonte e gere sequência de tokens;
Traduza sequência de tokens para AST;
Inclua informações semânticas na AST;
Gere código a partir da AST;

▪ **Opção 2 para o mesmo Compilador:**

while houver funções C do;
 Leia próxima função;
 Gere código para a função;
end;
Resolva referências cruzadas;



Top-Down X Funções do Sistema

❑ **Top-Down** supõe que todo sistema pode, no nível mais abstrato, ser descrito por uma única função.

❑ Função principal de um Sistema Operacional é:

Processar todas as requisições do usuário,

que pode ser refinada em: **while tudo em ordem do**
 leia pedido do usuário;
 ponha pedido na fila de entrada;
 retira um pedido r da fila ;
 processe o pedido r;
 ...
 end



Top-Down X Funções do Sistema

- ☐ **Top-Down** supõe que todo sistema pode, no nível mais abstrato, ser descrito por uma única função.
- ☐ Sistemas de software são mais bem descritos pelos **vários** serviços que oferecem.
- ☐ Sistema Operacional oferece serviços de:
 - gerência do processador
 - gerência de memória
 - entrada e saída
 - interpretação de comandos



... Top-Down X Funções do Sistema

- ☐ Sistemas de software são mais bem descritos pelos **vários** serviços que oferecem.
- ☐ Sistema Operacional oferece serviços de:
 - gerência do processador
 - gerência de memória
 - entrada e saída
 - interpretação de comandos



Top-Down X Estruturas de Dados

- ☐ Foco de atenção centrado na função prejudica a influência da estrutura de dados na arquitetura: Vide pesquisa em tabela
- ☐ Top-down traz *conveniência a curto prazo e inflexibilidade a longo prazo*.
- ☐ Top-down é ainda útil para pequenos programas e algoritmos individuais.
- ☐ Estruturas de dados mais estáveis do que funções.
- ☐ Foco de atenção inteiramente voltado para estrutura de dados também pode não ser a solução.
- ☐ **Tipos abstratos de dados** provêm o equilíbrio necessário.
- ☐ Dado como elemento principal de modularização favorece:
 - **Reusabilidade**
 - **Extensibilidade**



Fatos

- ☐ Necessidade de técnicas de organização de software que privilegiem:
 - **extensibilidade**
 - **reusabilidade**
- ☐ Necessidade de técnicas que suportem o desenvolvimento sistemático de software de forma a garantir:
 - **correção**
 - **robustez**
- ☐ Fatores externos são o que interessa, mas só podem ser atingidos por meio dos fatores internos:
 - MODULARIDADE**
 - ORIENTAÇÃO POR OBJETOS**