

Programação Orientada a Objetos

Universidade Federal de Uberlândia

Prof. Fabiano Dorça

Introdução

- O que é um paradigma?

É uma forma de abordar um problema, segundo um conjunto de procedimentos, valores ou conceitos que direcionam o pensamento.

Introdução

O paradigma de programação procedimental.

- ◆ Ausência de uma metodologia
- ◆ Atividade de programar muito peculiar a cada programador.

Gerando...

Uma insegurança na qualidade dos softwares produzidos.

Introdução

- ◆ Resolução de problemas através da definição de funções.
- ◆ Funções resultantes da decomposição do problema inicial em sub-problemas menores.

Introdução

Este processo:

- ◆ é chamado de Decomposição Funcional;
- ◆ identifica-se as principais funções ou passos de um processo
- ◆ que por sua vez podem ser decompostos em funções mais simples
- ◆ degrada o domínio do problema por não levar para a solução os conceitos envolvidos e suas relações.

Introdução

Vantagem:

- ◆ Aparantemente mais intuitivo e direto.

Desvantagens:

- ◆ Descreve apenas a solução para um problema, e não o problema
- ◆ Dificulta o entendimento da solução
- ◆ Dificulta a manutenção do software
- ◆ Dificulta a reutilização do software
- ◆ Quanto maior o problema, mais difícil fica.

Introdução

Prog. Procedimental:

- ◆ Foco na função (parte ativa do programa).
- ◆ Os dados são manuseados de forma passiva, perdendo sua importância no contexto.
- ◆ Perda da semântica do mundo real (seus conceitos e inter-relação entre eles) → GAP SEMÂNTICO
- ◆ Maior dificuldade na reutilização de código, comumente levando à cópia-cola.

Introdução

Prog. Procedimental:

- ◆ Dispersão de dados e funções, sem vínculo.
- ◆ **Dados** podem ser **lidos e modificados** por qualquer função, **desordenadamente**.
- ◆ **Não** existe um **modelo de dados**.
- ◆ A **construção de um modelo de dados** posteriormente é **mais confusa**.
- ◆ Dificuldade na depuração e correção.

Introdução

- ◆ A **Programação Orientada a Objetos (POO)**, foi criada para tentar aproximar o mundo real e o mundo virtual
- ◆ A ideia fundamental é tentar **simular o mundo real** dentro do computador.
- ◆ Para isso, nada mais natural do que utilizar objetos, afinal, nosso mundo é composto de objetos

Introdução

- ◆ Na POO, o programador é responsável por **moldar o mundo dos objetos**, e definir como os objetos devem **interagir entre si**.
- ◆ Isso possibilita a criação de códigos com **baixo acoplamento** e que podem ser facilmente reutilizados, o que são alguns dos principais motivos para se programar orientado a objetos.

Introdução

Prog. Orientada a Objetos:

- ◆ Os dados e as funções são vistos de forma agregada.
- ◆ Já existe um modelo de dados inerente ao programa.
- ◆ A semântica do mundo real é totalmente retratada na solução computacional OO.
- ◆ Unificação semântica entre as fases de análise, projeto e implementação.

Introdução

- ◆ Reutilizar conceitos do mundo real é muito mais fácil do que reutilizar funções e dados dispersos no programa.
- ◆ POO...
 - ◆ se apóia na estrutura fundamental do domínio da aplicação,
 - ◆ ao invés de se apoiar nos requisitos funcionais do problema.
- ◆ Estas características conduzem a uma maior facilidade na extensão do programa.

Introdução

Desta forma, o paradigma orientado a objetos visa:

- manutenibilidade: facilidade de manutenção corretiva e evolutiva.
- reusabilidade: facilidade em se desenvolver novos módulos de um software a partir de módulos já existentes.

Introdução

- ◆ IMPORTANTE:
- ◆ Os requisitos sempre mudam.
- ◆ Por isso, é importante preparar o sistema para mudanças futuras.
- ◆ O paradigma procedimental não foi concebido com esta preocupação (software imutável).

Introdução

- ◆ A POO resolve problemas importantes do processo de desenvolvimento de software:
 - ◆ Aumenta a produtividade.
 - ◆ Possibilita o reuso em larga escala.
 - ◆ Facilita a correção e evolução do software.
 - ◆ Diminui a taxa de erros.
 - ◆ Aumenta a qualidade do produto final.

Introdução

- ◆ O termo Programação Orientada a Objetos (POO) foi criado por **Alan Kay**, autor da linguagem de programação Smalltalk (70's).
- ◆ Algumas das idéias da POO já eram aplicadas, - linguagem Simula 67, criada por Ole-Johan Dahl e Kristen Nygaard em 1967.

Introdução

- ◆ Entretanto a POO só veio a ser aceito realmente nas empresas por volta dos anos 90.
- ◆ Alguns exemplos de linguagens orientadas a objetos: Java, C#, C++, Object Pascal (Delphi), Objective C, Ruby, Python, Lisp, PHP.

Referências

- BOOCH, G. Object-Oriented Analysis and Design with Applications, 3a Edição. Addison-Wesley, 2007.
- BOOCH, G., RUMBAUGH, J., JACOBSON, I. UML, Guia do Usuário. Rio de Janeiro: Campus, 2000.
- DEITEL, H. M.; DEITEL P. J. Java: Como Programar, 6a. Edição. Pearson, 2005. (Livro Texto)
- FOWLER, M. UML Essencial, 2a Edição. Bookmann, 2000.
- HORSTMANN, C.; CORNELL, G. Core Java 2 - Fundamentals, 7a. Edição. Prentice Hall, 2004.
- LARMAN, C. Utilizando UML e Padrões: Uma Introdução à Análise e ao Projeto Orientado a Objetos. Porto Alegre: Bookmann, 2001.
- RUMBAUGH, J.; BLAHA, M. Modelagem e Projetos Baseados em Objetos com UML 2, 1a Edição. Editora Campus, 2006