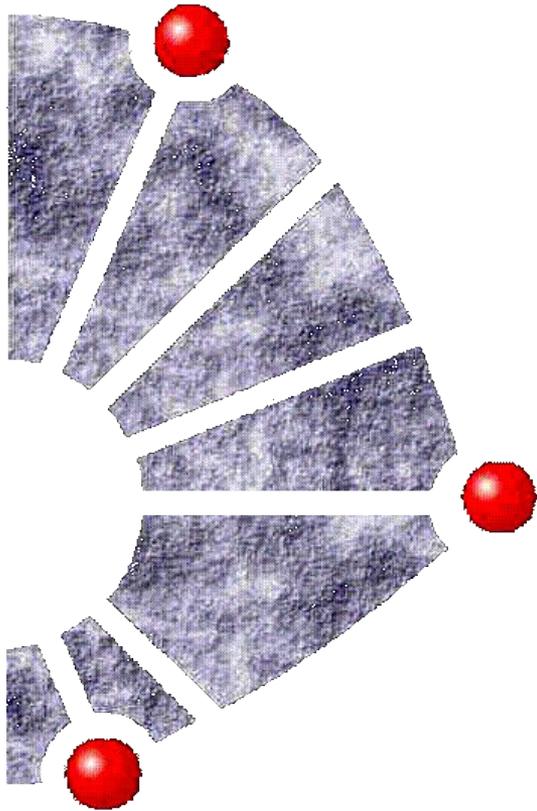


Programação Orientada a Objetos



FEEC-UNICAMP

Ricardo Gudwin



Programação Orientada a Objetos

- Metodologia de Programação Tradicional
 - Programação Estruturada
 - Sequência de comandos ou “procedimentos”
 - seleção
 - repetição
 - Fluxo de Controle
 - Sequência de Execução
- Programação Orientada a Objetos
 - Quebra de paradigma - objetos provendo serviços
 - atribuição de responsabilidades a objetos
 - Orientação a objetos
 - técnicas para a modelagem de sistemas, incluindo diversos conceitos, para esse propósito



Programação Orientada a Objetos

- Sistema
 - modelado por um conjunto de objetos que interagem entre si
- Objeto
 - elemento primitivo semanticamente relacionado ao mundo real sendo modelado
- Elementos de um Modelo de Objeto
 - Abstração (características essenciais - atributos e métodos)
 - Encapsulamento (ocultamento de informações, interface)
 - Modularidade (decomposição, coesão e acoplamento)
 - Hierarquia (níveis de abstração)
- Classe
 - conjunto de objetos com mesma estrutura e comportamento
 - objeto = instância de uma classe



Programação Orientada a Objetos

- Objeto
 - possui um estado, um comportamento e uma identidade
- Estado
 - todas as propriedades e valores correntes de propriedades
- Comportamento
 - como os objetos reagem = mudança de estados
 - conjunto de métodos
- Identidade
 - propriedade que distingue um objeto de outros de uma mesma classe
 - identificadores de instâncias de objetos



Programação Orientada a Objetos

- Métodos Especiais
 - construtor e destrutor
- Herança
 - mecanismo que permite a definição de um objeto a partir de outros em uma hierarquia
 - simples ou múltipla
 - mecanismo útil para a reutilização de software
 - superclasse e subclasse
- Polimorfismo
 - mesmo nome de método para diferentes classes
- Overriding
 - redefinição de um método já definido em uma superclasse



Objetos Distribuídos

■ Objetos Clássicos

- blob de inteligência encapsulando código e dados
- facilitam a reutilização de código
- “existem” somente dentro do escopo de programas isolados

■ Objetos Distribuídos

- blob de inteligência que pode existir em qualquer lugar da rede
- pedaços independentes de código que podem ser acessados por clientes remotos por invocação de métodos
- linguagens e compiladores utilizados nos servidores não influem em seu uso por clientes
- clientes não precisam saber onde estão os objetos e nem em que sistema operacional este executa



Por que Objetos ?

- Programação Orientada a Objetos causou um grande impacto na indústria de desenvolvimento de software
 - reutilização de código
 - altos níveis de produtividade no desenvolvimento de software
 - objetos podem simular, descrever e analisar entidades reais que compõem os processos ocorrendo dentro de uma organização comercial ou industrial
- Componentes
 - peças independentes de software que podem se interconectar através de redes, aplicações, linguagens, ferramentas e sistemas operacionais
 - objetos ou conjuntos de objetos com uma interface bem definida



Componentes Inteligentes ou Super-componentes

- Componentes Inteligentes
 - segurança, controle de licença, controle de versão, ciclo de vida, gerenciamento, suporte para palhetas em ferramentas abertas, notificação de eventos, configuração e gerenciamento de propriedades, scripting, meta-dados e introspeção, controle de transações, locking, persistência, relacionamentos, auto-teste, auto-instalação, etc...
- Business Objects: Os Componentes do Futuro
 - "business objects" fazem mais do que simplesmente cooperar - eles "colaboram" de modo a realizar suas tarefas
 - autonomia e graus de liberdade na tomada de decisões
 - agentes e agentes inteligentes