Modelagem e Programação Orientada a Objetos

BSI – DEINFO – UFRPE

2010.1

Missão

- Apresentar e exercitar conceitos de desenvolvimento de software orientado a objetos
- Destacar a importância de modelagem com UML (Linguagem de Modelagem Unificada)
- Motivar, apresentar, exercitar e consolidar o uso de técnicas de programação orientada a objeto que tenham um impacto considerável sobre qualidade de *software*.

Objetivos

- Discutir aspectos de qualidade e modularidade de *software*
- Introduzir conceitos de modelagem OO
- Introduzir conceitos de POO e Java
- Indicar como programas em Java podem ser adequadamente escritos e estruturados
- Utilizar ambientes de programação em Java
- Desenvolver uma aplicação de médio porte

Relevância e Motivação

- Técnicas a serem utilizadas na prática
- Desenvolver *software* de qualidade
- Java corresponde ao estado-da-arte
- Impacto econômico e social

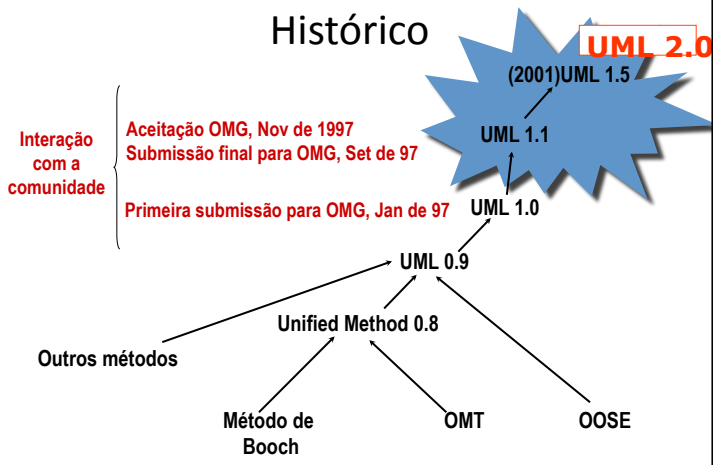
Clareza versus Eficiência

- Pensamento a longo prazo e elegância, ao invés de imediatismo e resultados de qualquer jeito
- *Software* tem que ser adaptável, flexível, fácil de mudar (custos baixos, mudanças rápidas)

Unified Modeling Language (UML)

- Precisamos de uma linguagem para expressar modelos
  - Vocabulário e regras
  - Permite visões variadas (decomposição), abstratas, conectadas entre si
- UML (1997): Padrão para "plantas" de software

Histórico



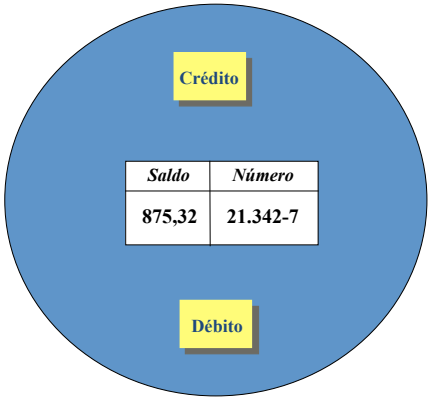
UML é uma Linguagem

- Padrão para "plantas" de software
  - Mas pode servir para outras representações
- Oferece múltiplas perspectivas de um problema
- Não definem como o desenvolvimento será realizado
  - Esse é o papel de um processo (métodos)

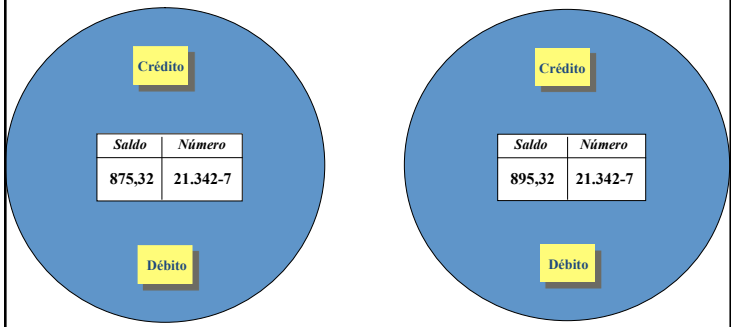
Programação Orientada a Objetos

- Foco nos dados (objetos) do sistema, não nas funções
- Estruturação do programa é baseada nos dados, não nas funções
- As funções mudam mais do que os dados

Objeto Conta Bancária



Estados do Objeto Conta



O que é Java?

"Java é uma linguagem simples, orientada a objetos, distribuída, interpretada, robusta, segura, independente de arquitetura, portável, de alto desempenho, concorrente e dinâmica"

Implementações de Java

- Interpretada
  - *Bytecodes* da JVM
  - Independente de plataforma
- Compilada
  - Código nativo em C ou C++
- Alto desempenho?

Alta Performance?

- Java oferece alternativas, compromissos
- Código do cliente pode ser interpretado
  - compiladores JIT aumentam desempenho
  - cliente universal
  - código móvel, carregado pela rede, evitando instalações, conflito entre versões, etc.
- Código do servidor pode ser compilado para código de máquina
  - desempenho similar a C++

Java é Simples

- Sintaxe familiar a vários programadores (baseada em C e C++)
- Elimina várias redundâncias de C++
- Simples para algumas aplicações, desde que se conheça alguns pacotes
- Simples, dado tudo que a linguagem oferece

Java eliminou...

- Ponteiros
- *goto, struct* e *union*
- Número variável de argumentos
- Tipos fracos
- Criação e remoção de objetos
  - alocar e liberar memória explicitamente

Java é Orientada a Objetos

- Objetos e Classes
- Encapsulamento de dados e operações
- Subtipos e Herança
- Polimorfismo
- Ligações dinâmicas (*dynamic binding*)
- Criação e remoção dinâmica de objetos
  - alocação e liberação automática de memória

Java é Portável

- Em tese, redução de custos com migração, instalação, treinamento, etc.
- Na prática, ainda é necessário depurar programas antes de migrar para outra plataforma (awt)
- Mas toda a arquitetura está pronta (swing)

Java é Distribuída

- Oferece suporte de alto nível para acesso a Internet/WWW (pacote java.net)
- Objetos distribuídos com RMI e CORBA
- Suporte para acesso a arquivos remotos, banco de dados, etc.

Java é Robusta

- Ausência de ponteiros
- Fortemente tipada
- Coleta de lixo automática
- Exceções tipadas
- Acesso a *arrays* é verificado
- Variáveis são inicializadas automaticamente
  - com exceção de variáveis locais de métodos que devem ser inicializadas explicitamente

Java é Concorrente

- Essencial para implementar interfaces gráficas decentemente
- Métodos sincronizados
- Monitores

Java: linguagem e ambiente

- Acesso a Internet e WWW (java.net)
- Applets (java.applet)
- Definição de interfaces gráficas (java.awt)
- Suporte a objetos distribuídos (java.rmi)
- Interface com Banco de Dados (java.sql)
- Básicos: *threads* e exceções (java.lang), arquivos (java.io), utilitários de propósito geral (java.util)

Referências

- UML – Guia do Usuário. BOOCH, G., RUMBAUGH, J., JACOBSON, I. Ed. Campus 2 ed. 2006.
- Java: Como Programar. Harvey Deitel e Paul Deitel, 6a edição, Prentice Hall, 2005.
- Site de Java da SUN, http://java.sun.com/

Avaliação

- 1 VA
  - Exame escrito
- 2 VA
  - 50% exame escrito
  - 50% projeto
    - 50% das versões 1 e 2 (25% cada)
    - 50% da versão 3
    - a pontualidade na entrega das etapas do projeto é um fator de avaliação
    - 10% de penalidade por aula de atraso

Agradecimento

- Slides parcialmente preparados a partir de material cedido pelo prof. Sérgio Soares