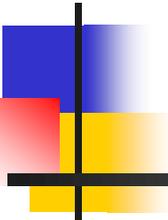


Programação Orientada a Objetos



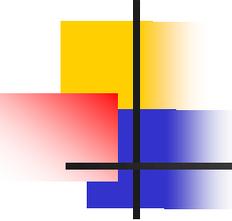
Universidade Católica de Pernambuco
Ciência da Computação

Prof. Márcio Bueno
poonoite@marciobueno.com

Fonte: Material da Prof^a Karina Oliveira

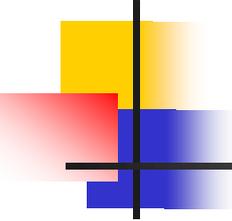
Introdução ao Paradigma OO

Prof. Márcio Bueno
poonoite@marciobueno.com



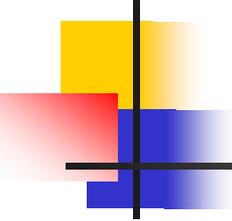
Conceito

- Paradigma que usa ao longo do processo de desenvolvimento de software, o conceito de "Objetos".
- Sistemas OO são estruturados como um conjunto de objetos inter-relacionados que juntos atendem os requisitos do sistema.



Conceito (2)

- Programação imperativa versus Programação OO
 - A programação imperativa primeiro focaliza as funções que um sistema vai oferecer ao usuário;
 - A programação orientada a objetos dá prioridade a definição dos objetos (dados) a serem manipulados pelo sistema e em seguida as atividades que um dado objeto poderá realizar.



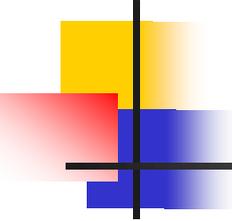
Vantagens e Desvantagens

■ Vantagens

- Modelagem usando conceitos do mundo real;
- Transição suave entre etapas de desenvolvimento devido ao uso da abstração de objetos;
- Grande suporte de ferramentas, linguagens e tecnologias, métodos e técnicas.

■ Desvantagens

- Uso correto requer experiência e tempo inicial para aprendizado.



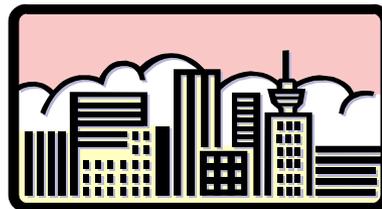
Princípios Básicos da POO

- Abstração
- Encapsulamento
- Modularidade
- Herança

Abstração

- Criação de um modelo para representação da realidade
- Representação apenas das características essenciais do ponto de vista do desenvolvimento do sistema
- Concentração em idéias gerais ("o que"), não em detalhes ("como")

Cachorro
nome
idade
raca



Imovel
precoAluguel
precoCompra
quantidadeQuartos
quantidadeSuites
tamanho

Galinha
idade



Encapsulamento

- Capacidade de esconder detalhes específicos de uma dada entidade
- Elimina dependência de implementação, escondendo-a de quem usa o objeto no sistema



Relogio
◆ horaAtual() ◆ corrigirHora() ◆ corrigirMinuto() ◆ corrigirSegundo() ◆ dataAtual() ◆ corrigirDiaSemana() ◆ corrigirMes() ◆ corrigirAno()

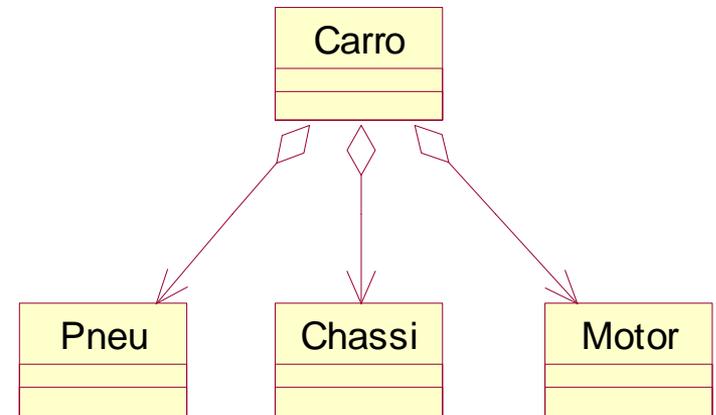
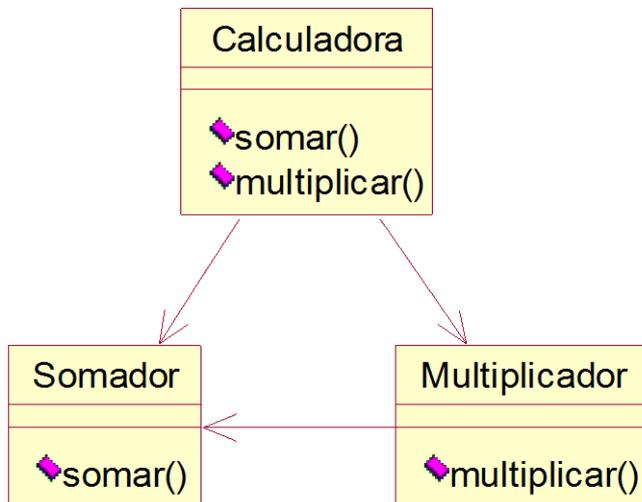


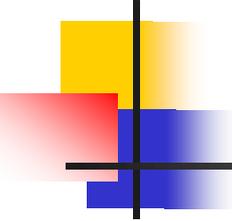
Televisao
◆ ligar() ◆ desligar() ◆ mudarCanal() ◆ aumentarVolume() ◆ reduzirVolume()



Modularidade

- Decomposição de alguma entidade grande e complexa em entidades menores
- Permite decompor um problema complexo em problemas menores e mais simples
- Sistema grande quebrado em unidades menores
- **Estratégia de dividir para conquistar!**



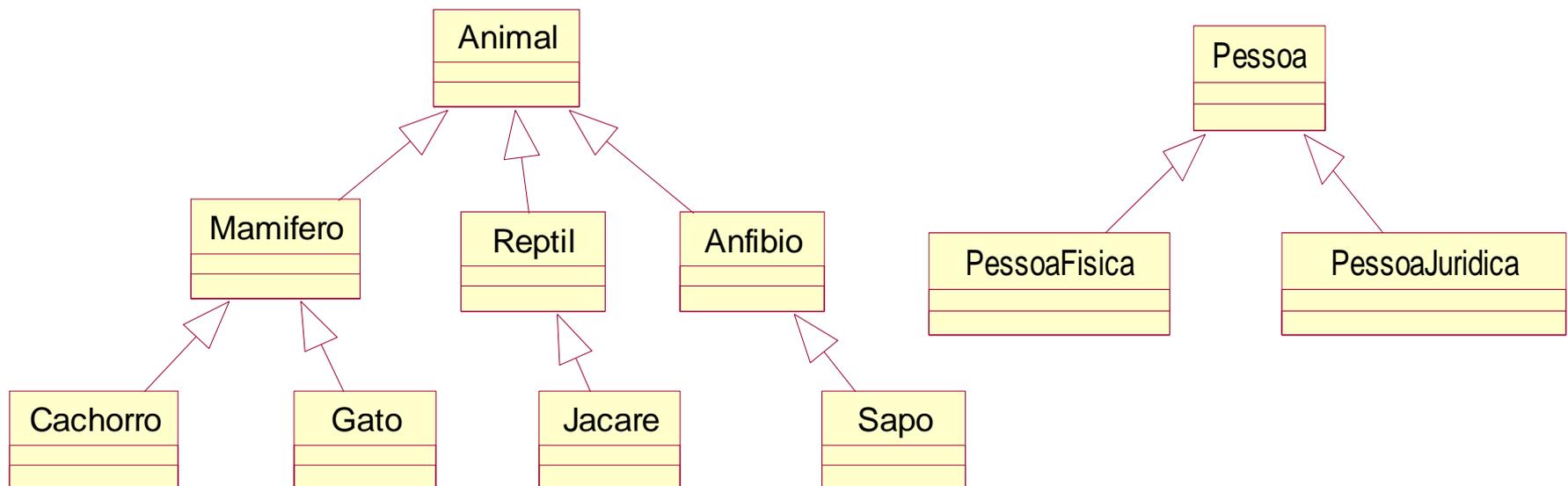


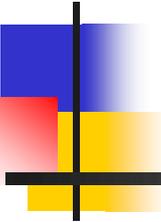
Modularidade (2)

- O método de construção de um software é modular quando ele ajuda ao projetista a produzir um sistema de software feito de elementos autônomos conectados por uma coerência e tenha uma estrutura simples

Herança

- Hierarquização de abstrações
- Mecanismos para permitir **reuso** e **extensibilidade**
- Relação do tipo "é um" ou relação de herança





Características da Linguagem Java

Prof. Márcio Bueno
poonoite@marciobueno.com

Características da Linguagem Java

- Concisa e Simples
- Orientada a Objetos
- Linguagem de programação para a Internet
- Robusta
- Segura
- Neutra em relação a Arquitetura
- Portável
- Interpretada
- Concorrente
- Tolerante à Falhas
- Coleta automática de lixo
- Compilada

Características da Linguagem Java

■ Concisa e simples

- Sintaxe parecida com C o que facilita o entendimento por uma grande parte de programadores
- Não suporta manipulação de ponteiros
- Não utiliza alocação explícita de memória
- Não tem herança múltipla

■ Orientada a objetos

- Suporta os principais conceitos de orientação a objetos
- Favorece a reusabilidade e extensibilidade

Características da Linguagem Java (2)

- Linguagem de programação para a Internet
 - Poderosos recursos de rede;
 - Biblioteca extensa de rotinas para lidar com protocolos TCP/IP, como HTTP e FTP;
 - Acesso a objetos na Internet através de URLs;
 - Mecanismos de invocação de métodos remotos.
- Robusta
 - Fortemente tipada → Aumenta a confiabilidade dos programas
 - Verificação em tempo de compilação e execução
 - Reduz imprevistos em tempo de execução
 - Ex.: Atributos são automaticamente inicializados.

Características da Linguagem Java (3)

■ Segura

- Eliminação dos principais problemas de segurança como restrições de acesso a arquivos
- Criptografia / Assinatura Digital

■ Neutra em relação à Arquitetura

- Compilador gera formato de arquivo neutro em relação à arquitetura
- Código compilado é executável em muitos processadores, dada a presença do sistema Java de tempo de execução
- Geração de bytecodes que são posteriormente interpretados em qualquer máquina e convertidos para código nativo de máquina

Características da Linguagem Java (3)

■ Portável

- Completamente especificada
- Não contém aspectos dependentes da implementação: o tamanho dos tipos é fixo para qualquer implementação.
- "*Write Once, Run Anywhere*" (SUN)

Características da Linguagem Java (3)

■ Interpretada

- Interpretador Java pode executar bytecodes Java diretamente em qualquer máquina na qual o interpretador tenha sido escrito.
- Facilita distribuição de software (classes são "baixadas" pela rede).

Características da Linguagem Java (3)

■ Concorrente

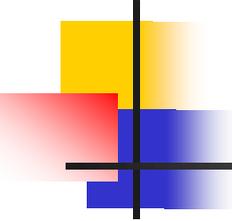
- Suporta aplicações concorrentes: *multithreads*
 - Uma aplicação pode ter diversas atividades "simultaneamente" (um Web browser pode estar emitindo o som de um clip, enquanto fazendo *scrolling* de uma página e ainda carregando uma imagem em *background*)
 - Importante para aplicações gráficas
- Melhor tempo de resposta para aplicações interativas e/ou de tempo-real.

Características da Linguagem Java (4)

- Tolerância a falhas através de exceções
 - Exceção - evento que ocorre durante a execução de um programa e que quebra o fluxo normal de execução das instruções
 - Java suporta o lançamento e a captura de exceções nas aplicações
 - Novas exceções podem ser criadas para tratar aspectos específicos
 - Padronização e legibilidade de código.

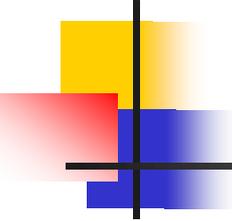
Características da Linguagem Java (4)

- Coleta automática de lixo
 - Evita erros de uso de memória
- Compilada - Alto Desempenho
 - Utilizando compiladores JIT (*Just-In-Time*), bytecodes podem ser traduzidos em tempo de execução para código de máquina.
- Extenso conjunto de bibliotecas de classes já desenvolvidas e incorporadas



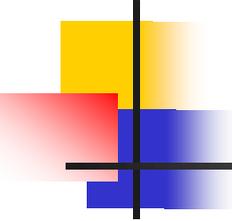
Tecnologia Java

- Uma coleção de APIs (Application Programming Interfaces)
- Um ambiente de desenvolvimento J2SDK
 - Java 2 Software Development Kit
 - J2SE - Standard Edition
 - J2EE - Enterprise Edition
 - J2ME - Micro Edition.
 - Coleção de ferramentas de linha de comando para, entre outras tarefas, compilar e executar aplicações Java.



Tecnologia Java

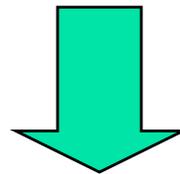
- Um ambiente de execução disponível para vários sistemas operacionais e browsers.
 - JRE - Java Runtime Environment
 - Tudo o que é necessário para executar aplicações Java.
- Compilação e execução
 - Um programa escrito usando a Linguagem Java é traduzido em **bytecode** através do processo de compilação e armazenado em um arquivo *.class chamado de Classe Java
 - **Bytecode** é o código de máquina que executa em qualquer S.O. através da Máquina Virtual Java (JVM)



Compilação e Execução

Código Java: **HelloWorld.java**

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello world!");  
    }  
}
```



compilação: **javac**

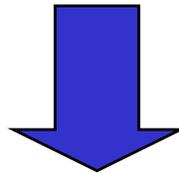
Bytecode Java: **HelloWorld.class**

```
F4 D9 00 03 0A B2 FE FF FF 09 02 01 01 2E 2F
```

Compilação e Execução

Bytecode Java: `HelloWorld.class`

F4 D9 00 03 0A B2 FE FF FF 09 02 01 01 2E 2F



execução: `java`

