Universidade Federal do Ceará
Centro de Ciências/Departamento de Computação
Código da Disciplina: CK0084
Professor: Ismayle de Sousa Santos

Aula 15, 16 e 17

Sistemas de Informações e Banco de Dados

Introdução à Programação Orientada a Objetos e Introdução ao Java

IsmayleSantos
ismayle@ufc.br
@IsmayleSantos

Agenda

- Introdução à Programação Orientada a Objetos
  - Programação Estruturada vs Orientada a Objetos
  - Origem
  - Conceito
  - Objetos - Atributos e Métodos
  - Classes
  - Encapsulamento
  - Visibilidade

O que é Programação Estruturada?

- E uma programação desenvolvida por Michael A. Jackson no livro "Principles of Program Design" de 1975
- Exemplo
  - PHP, Cobol, C
- Essa programação preconiza que todos os programas possíveis podem ser reduzidos a apenas três estruturas:
  - Sequência
  - Decisão e
  - Iteração (repetição)

O que é Programação Estruturada?

- Sequência: Uma tarefa é executada após a outra, linearmente
- Decisão: A partir de um teste lógico, determinado trecho de código é executado, ou não
- Iteração: A partir de um teste lógico, determinado trecho de código é repetido por um número finito de vezes

Um programa é tipicamente escrito em uma única função

Quais as Vantagens e Desvantagens da Programação Estruturada?

- Vantagens
  - É fácil de entender!
  - Muito usada em cursos introdutórios de programação
  - Execução mais rápida
- Desvantagens
  - Baixa reutilização de código
  - Códigos confusos com dados misturados com comportamento

O que é Programação Orientada a Objetos?

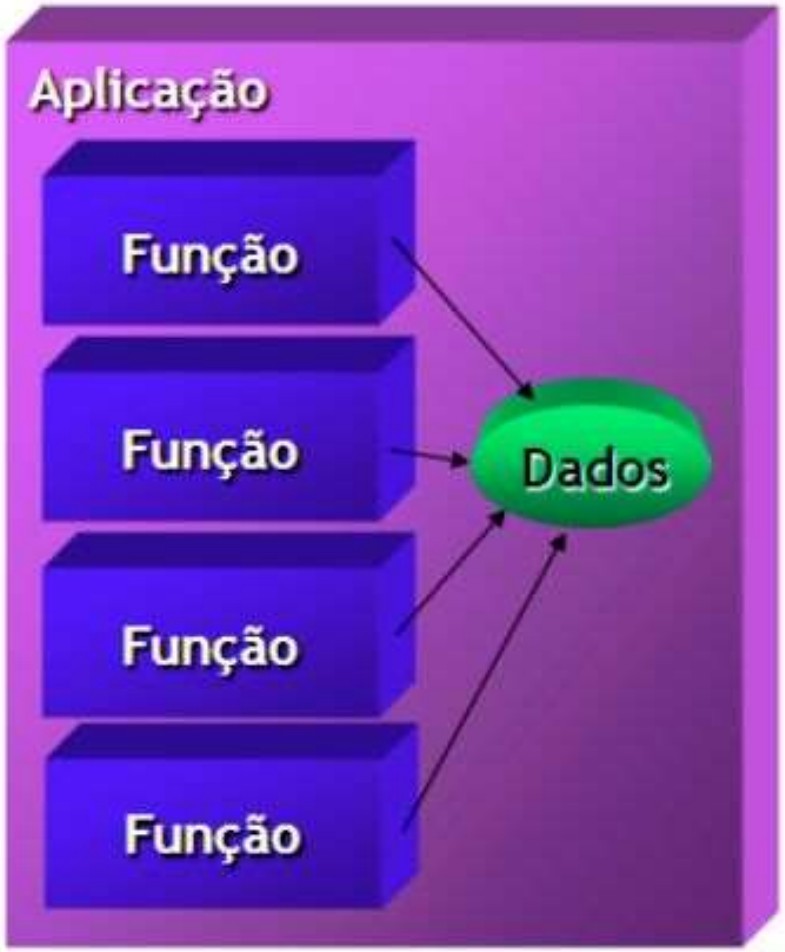
- A programação orientada a objetos (POO) é um modelo de programação onde diversas classes possuem características que definem um objeto

Quais as Vantagens e Desvantagens da POO?

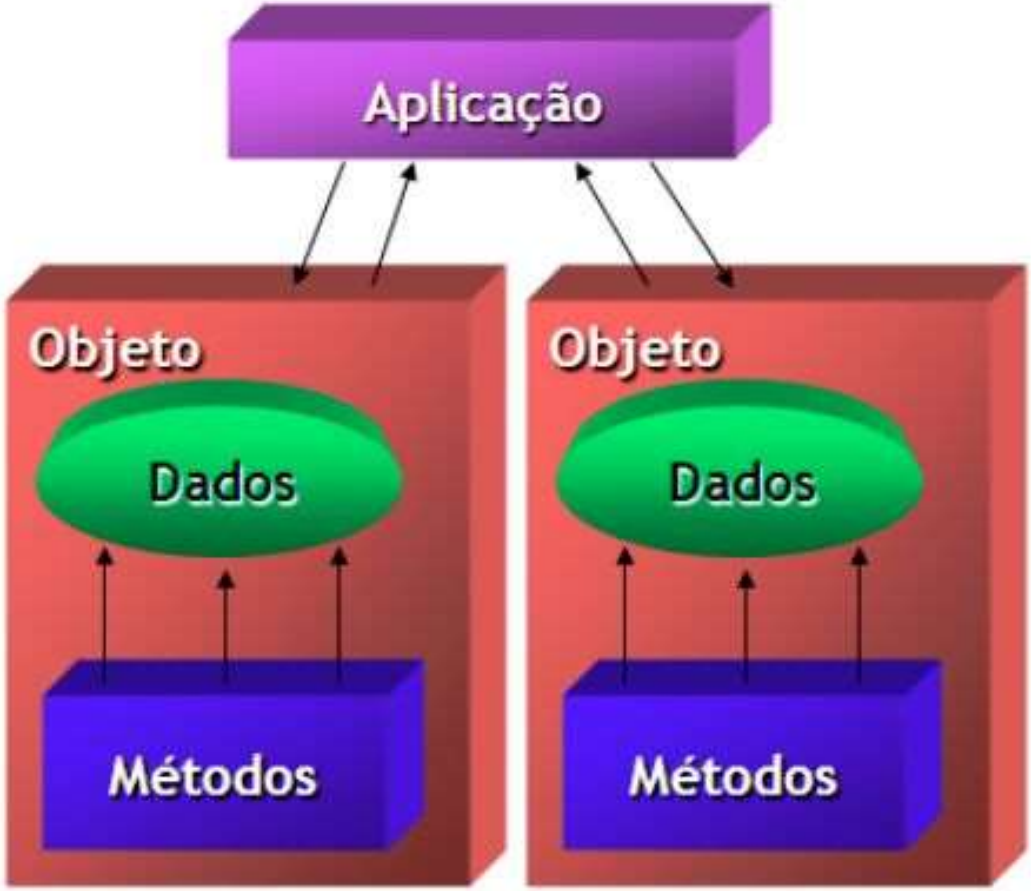
- Vantagens
  - Melhor organização do código
  - Bom reaproveitamento de código
- Desvantagens
  - Desempenho mais baixo que o paradigma estruturado
  - (Pode ser) Mais difícil a compreensão

Programação Estruturada vs POO

ESTRUTURADA



ORIENTAÇÃO A OBJETOS

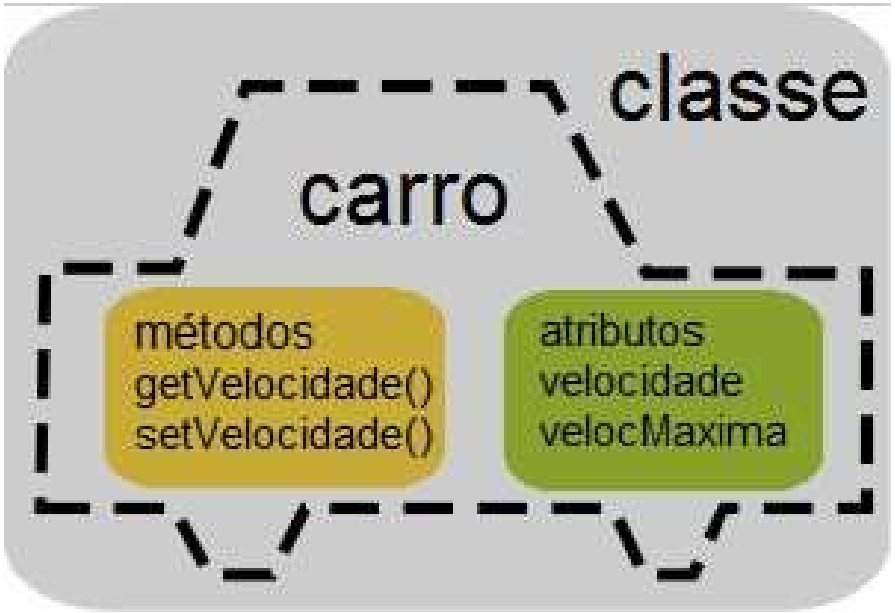


Qual a Origem da POO?

- Nos anos 70 surge Smalltalk, a primeira linguagem totalmente em Orientação a Objeto (O.O)
- C++, evolução de C, já possuía conceitos O.O
- Na década de 80 praticamente todas as linguagens já usavam conceitos O.O
  - Delphi
  - PASCAL
  - Java

Qual o Conceito por trás da POO?

- James E. Rumbaugh afirma que a POO é "Uma nova maneira de pensar os problemas utilizando conceitos do Mundo Real. [...] O componente fundamental é o objeto que combina estrutura e comportamento em uma única entidade"



O que é uma POO?

- Paradigma de programação baseado no conceito de classes e objetos
  - As classes são elementos onde dados e procedimentos são agrupados, segundo seu objetivo, para um determinado sistema
  - Quando uma classe é usada como um tipo de dado para a criação de uma variável, esta é chamada de objeto

O que a POO Engloba?

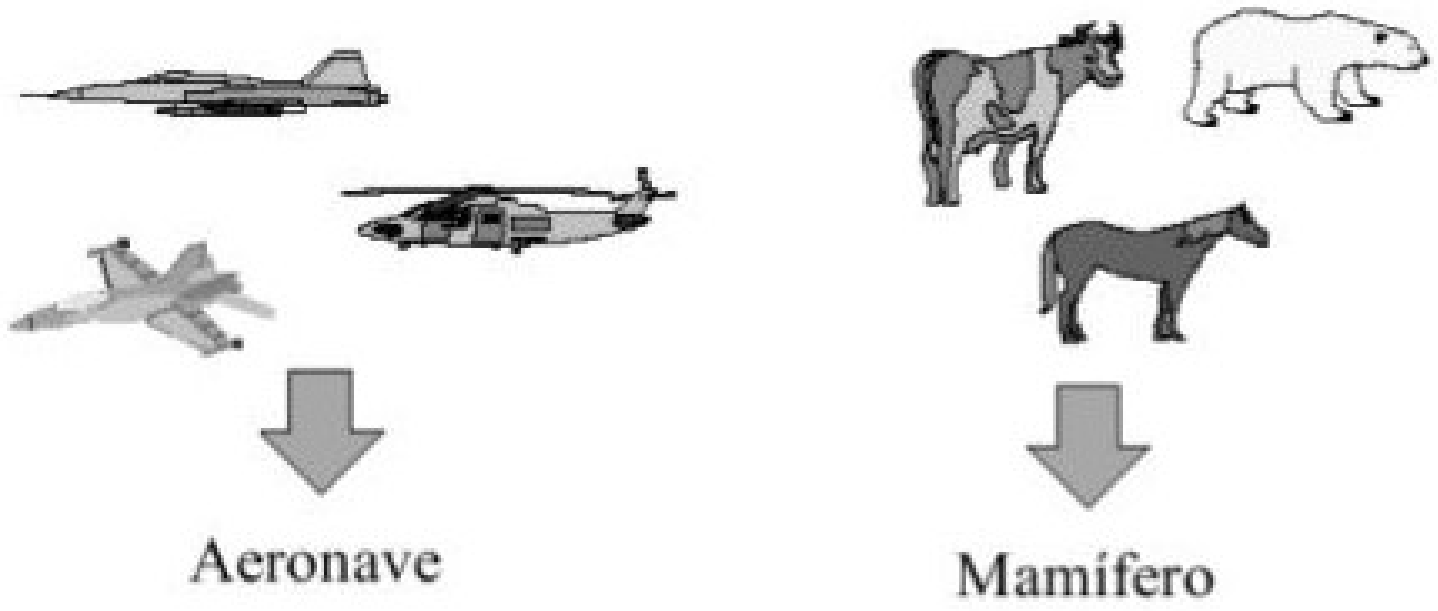
O que é o Princípio da Abstração?

- Habilidade de se concentrar nos aspectos essenciais do sistema, ou um contexto qualquer, ignorando o que é supérfluo
- Ou o processo de identificar os aspectos essenciais de um contexto qualquer, ignorando características menos importantes
  - A abstração é o resultado desse processo
- A abstração transforma aquilo que observamos no mundo real para a virtualidade

Classificar é uma forma de Abstração

- A abstração deve ser feita com algum objetivo para determinar o que é e o que não é importante

EXEMPLO:



O que é um Objeto?

- A percepção dos seres humanos é dada através dos objetos
  - Um objeto é uma entidade que exibe algum comportamento bem definido
- É a representação computacional de algo do mundo real
  - Concreto = pessoas, avião, carro ...
  - Abstrato = música, operação bancária

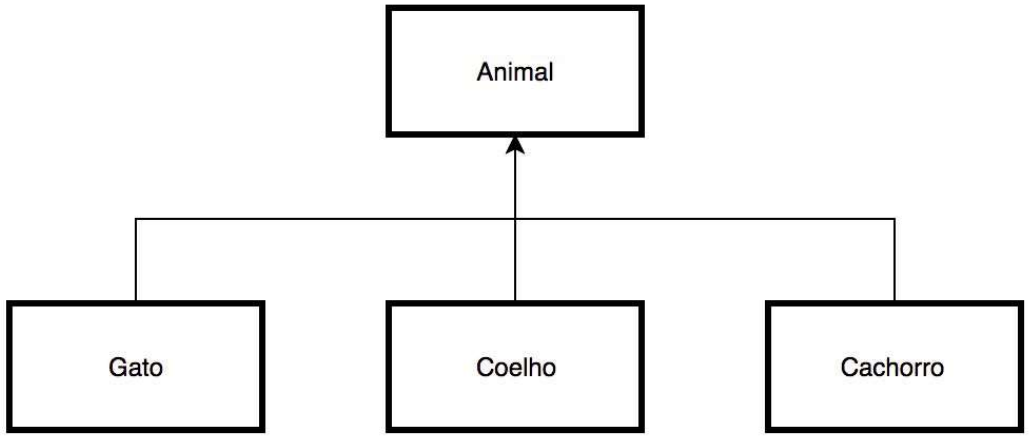
O que é um Objeto?

- Estado
  - Atributos (Características)
- Operações
  - Métodos (Comportamentos)
- Identidade
  - Dois objetos com estado e operações precisamente idênticos não são iguais
- Operações podem mudar os valores dos atributos assim mudando o estado de um objeto

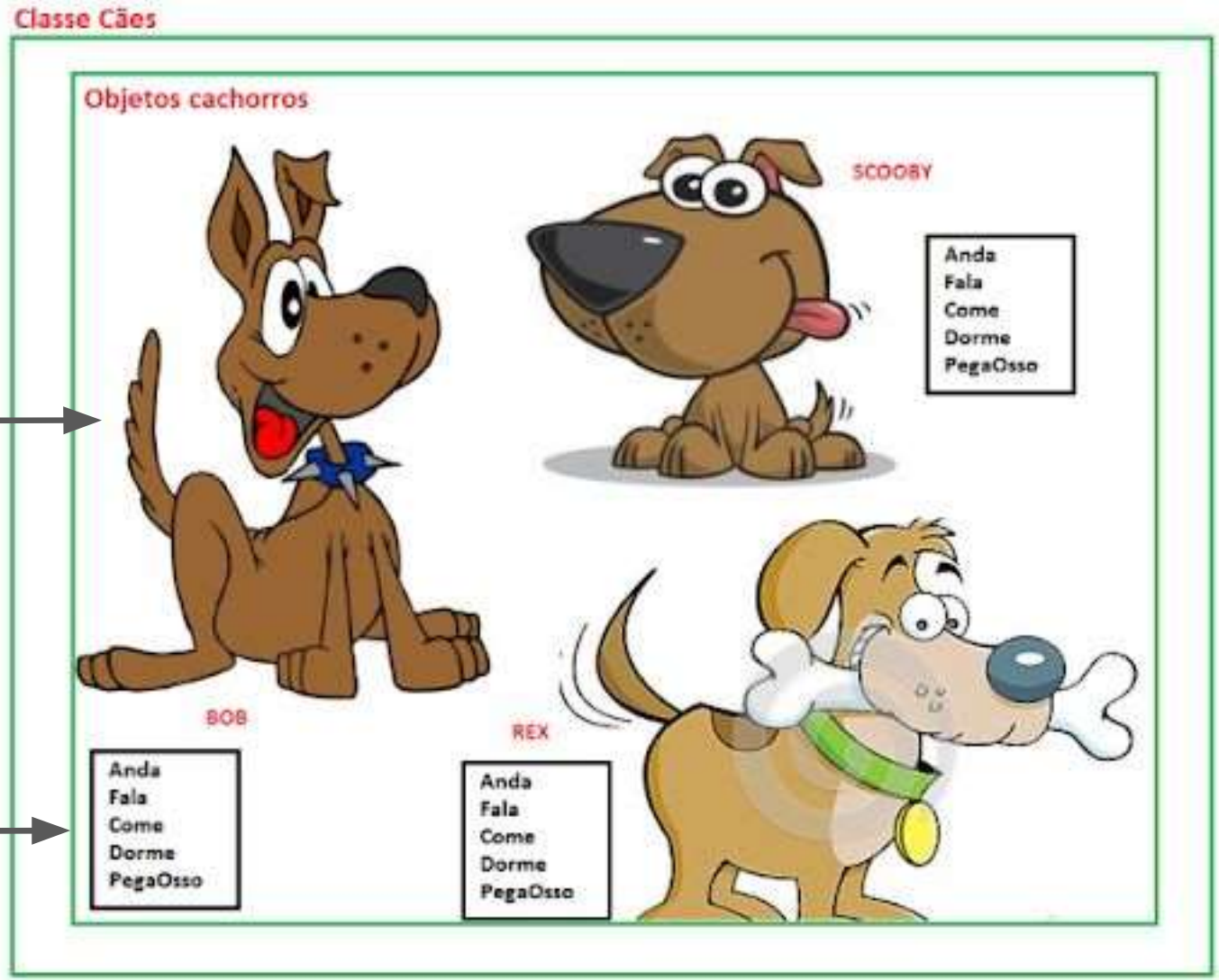
Exemplo de Objeto

Como Definir um Objeto?

- Objetos parecidos têm a mesma classificação
  - Carro x, cor azul, 2 portas;
  - Carro y, cor verde, 4 portas;
  - Ambos são classificados como carro
- O conhecimento a determinado objeto é dado a partir de sua classificação



Como Definir um Objeto?



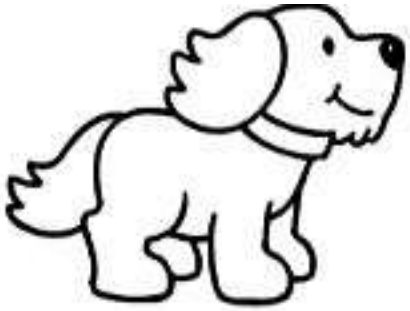
Atributo

Método

O que é um Método?

- São ações que uma classe possui

Exemplo de Métodos e Atributos



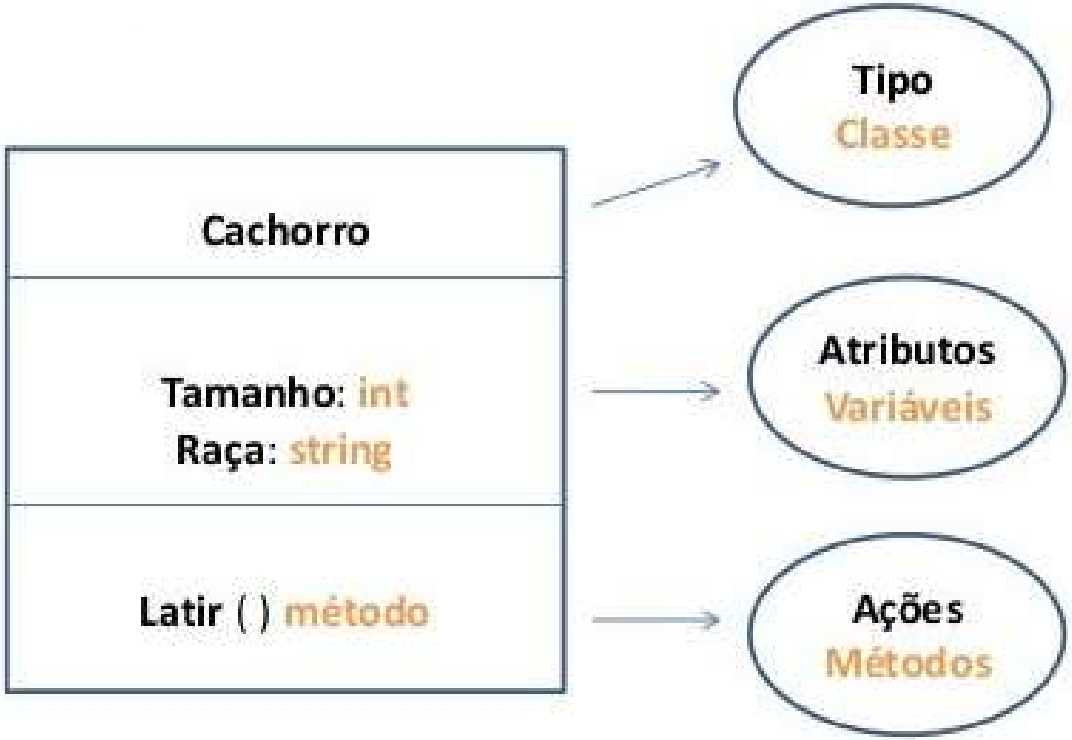
- Atributos
  - Raça: Poodle
  - Nome: Rex
  - Peso: 5 quilos
- Método
  - Latir
  - Comer
  - Dormir

- Potência: 500cc
- Modelo: Honda
- Ano: 1998
- Acelerar
- Frear
- Abastecer

O que é uma Classe?

- É um conjunto de objetos
  - Características semelhantes
  - Comportamento comum
  - Interação com outros objetos
- Uma classe é a forma para criação de objetos
- Objetos são representações concretas (instâncias) de uma classe
- Uma classe serve de modelo para vários objetos semelhantes que possuem os mesmos tipos de informação em seu estado e tem os mesmos comportamentos

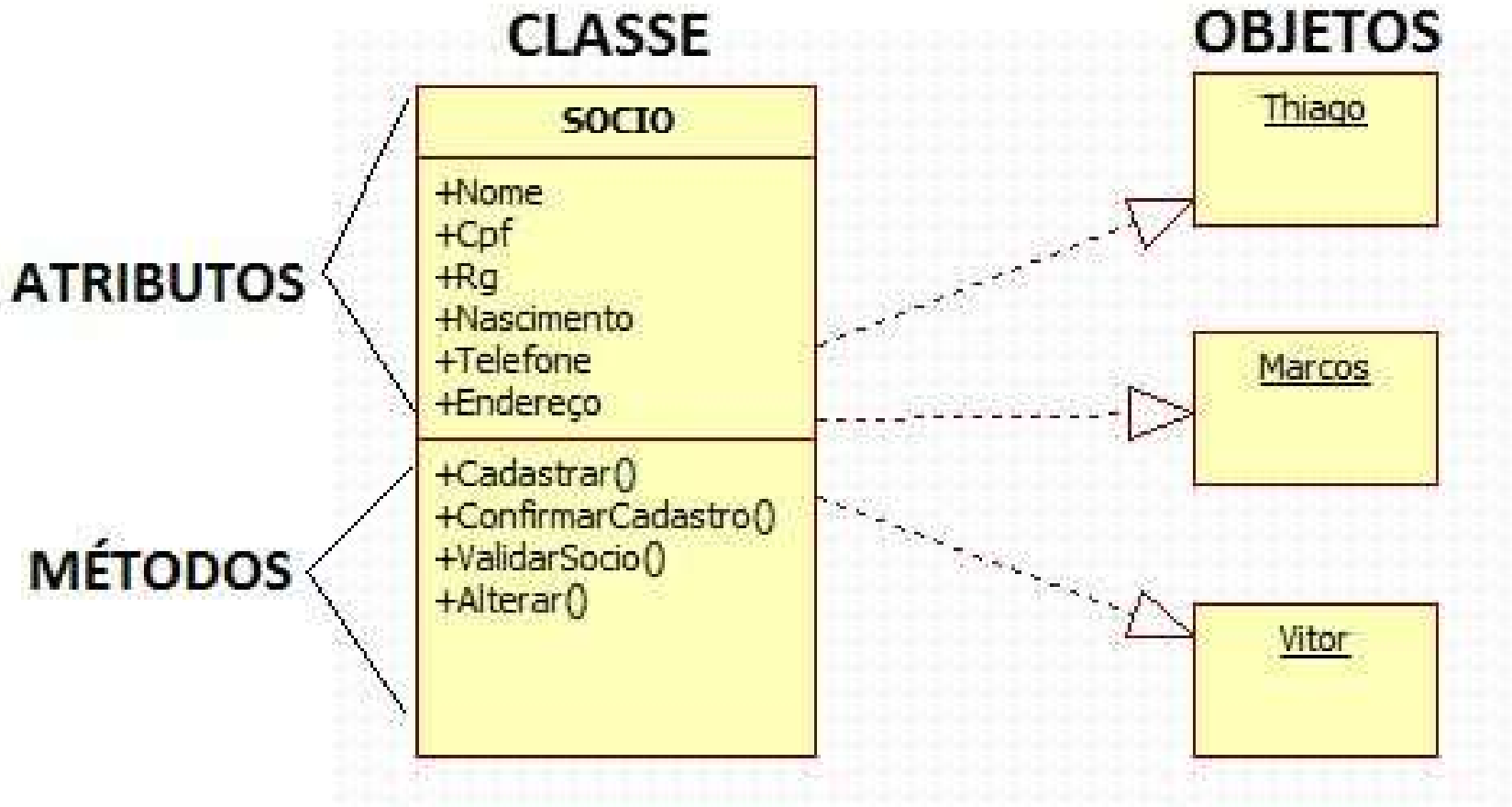
Exemplo de Classe



Sobre uma Classe ...

- É possível criar vários objetos em uma só classe
- Pode ser definido outros objetos com atributos diferentes comportamentos diferentes, mas do mesmo jeito não deixa de ser um objeto
- O conceito disso em orientação a objetos isso é chamado de código reuso, ou seja, reutilização de código
- Objetos trocam mensagem entre si, objetos trocam mensagem entre si e pode trocar atributo de outro objeto

Exemplo de Classe e Objeto



O que é Encapsulamento?

- Um objeto, em um programa, "encapsula" todo o seu estado e o comportamento
- Os dados e as operações são agrupados e a sua implementação é escondida, protegida dos usuários
- É a capacidade de restringir o acesso a elementos de uma classe utilizando qualificadores
  - Um qualificador ou modificador é uma palavra reservada que define a visibilidade de determinado atributo ou método

Exemplo de Encapsulamento

1) Classe Pessoa

```
public class Pessoa{
    private String nome;
    public String getNome(){
        return this.nome;
    }
    public void setNome(String nome){
        this.nome = nome;
    }
}
```

2) Classe Principal

```
public class Principal{
    public static void main(String[] args){
            Pessoa p = new Pessoa();
            p.setNome("Joaozinho");
            System.out.println("Nome :"  + p.getNome()  );
    }
}
```

Visibilidade

- Private (-)
  - Somente a classe tem acesso
  - Não é transmitido por herança
- Protected (#)
  - Visível em toda a classe de um pacote
  - Transmitido por herança
- Public (+)
  - Torna o membro acessível de fora da definição de classe
  - Visível irrestritamente

Visibilidade

ClasseCaneta
+ modelo + cor - ponta # carga # Tampada
+ escrever() + rabiscar() + pintar() - tampar() - destampar()

```
                Classe Caneta
Publico         Modelo: Caractere
Publico         Cor: Caractere
Privado         Ponta: Real
Protegido       Carga: Inteiro
protegido       Tampada: Logico
Publico         Metodo rabiscar()
                        se (tampada) então
                escreva ("ERRO")
                        senão
                escreva ("RABISCO")
                        fimse
                FimMetodo
Privado         Metodo tampar()
                tampada=verdadeiro
                FimMetodo
                FimClasse
```

Agora vamos falar de Java

Na aula de hoje iremos aprender sobre Tipos primitivos, estruturas de controle, métodos e argumentos

Entendendo a Linguagem Java

- Java
  - É uma linguagem de programação orientada a objetos desenvolvida na década de 90
    - A história dela começa em 1991, quando um grupo de empregados da Sun Microsystems iniciaram o Projeto Green para pequenos dispositivos eletrônicos de consumo, tais como o PDA (Personal Digital Assistant)

Entendendo a Linguagem Java

- Java é tanto compilada como interpretada:
  - O compilador transforma o programa fonte em bytecodes
    - Bytecodes são instruções compreendidas pela Máquina Virtual Java
  - A Máquina Virtual Java (JVM) é um interpretador, que transforma as instruções em linguagem de máquina (permitindo a execução do programa)
  - "Write once, run anywhere" - slogan criado pela Sun, para demonstrar a portabilidade da linguagem (graças aos bytecodes)

Entendendo a Linguagem Java

- Como plataforma, Java compreende uma JVM e uma API (application programming interface)
  - Programas podem ser executados como aplicações tradicionais ou em páginas web
  - Applications - são executados pelo sistema operacional e podem ser:
    - console applications: quando não apresentam saída gráfica, somente textual
    - windowed applications: criam e gerenciam múltiplas janelas, usam mecanismos de GUI (graphical user interface) para a programação

Entendendo a Linguagem Java

- Applets - são programas executados pelo navegador Web, através de uma JVM própria (interna)
  - A característica principal dos applets é a utilização da própria área da página como interface
  - Applets são executados em um ambiente restrito, oferecendo segurança

Características da Linguagem Java

- Concisa e simples
  - Não contém redundâncias e é fácil de entender, implementar e usar
- Orientada a objetos
  - Suporta os principais conceitos de orientação a objetos e favorece reusabilidade
  - A abordagem de OO permite o desenvolvimento de sistemas de uma forma mais natural
- Provê acesso a Internet/WWW
  - Contém bibliotecas especiais que possibilitam o trabalho com protocolos TCP/IP como HTTP e FTP

Características da Linguagem Java

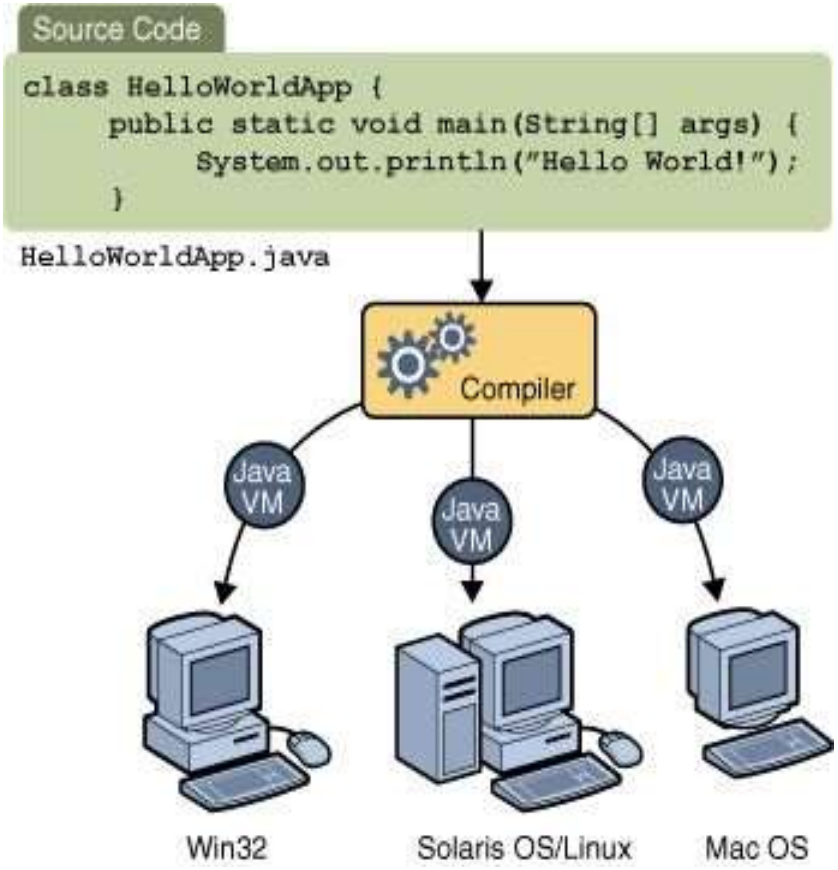
- Robusta
  - Reduz imprevistos em tempo de execução: variáveis são automaticamente inicializadas, uso disciplinado de ponteiros, rotinas devem ser chamadas corretamente, etc
- Portável
  - Não contém aspectos dependentes da implementação: o tamanho dos tipos é fixo para qualquer implementação, etc

Características da Linguagem Java

- Segura
  - Restrições de acesso a arquivos, manipulação de ponteiros, etc
- Concorrente
  - Suporta aplicações concorrentes: multithreads, monitores, execução atômica
- Também é: interpretada, neutra, portável, dinâmica e multi-thread

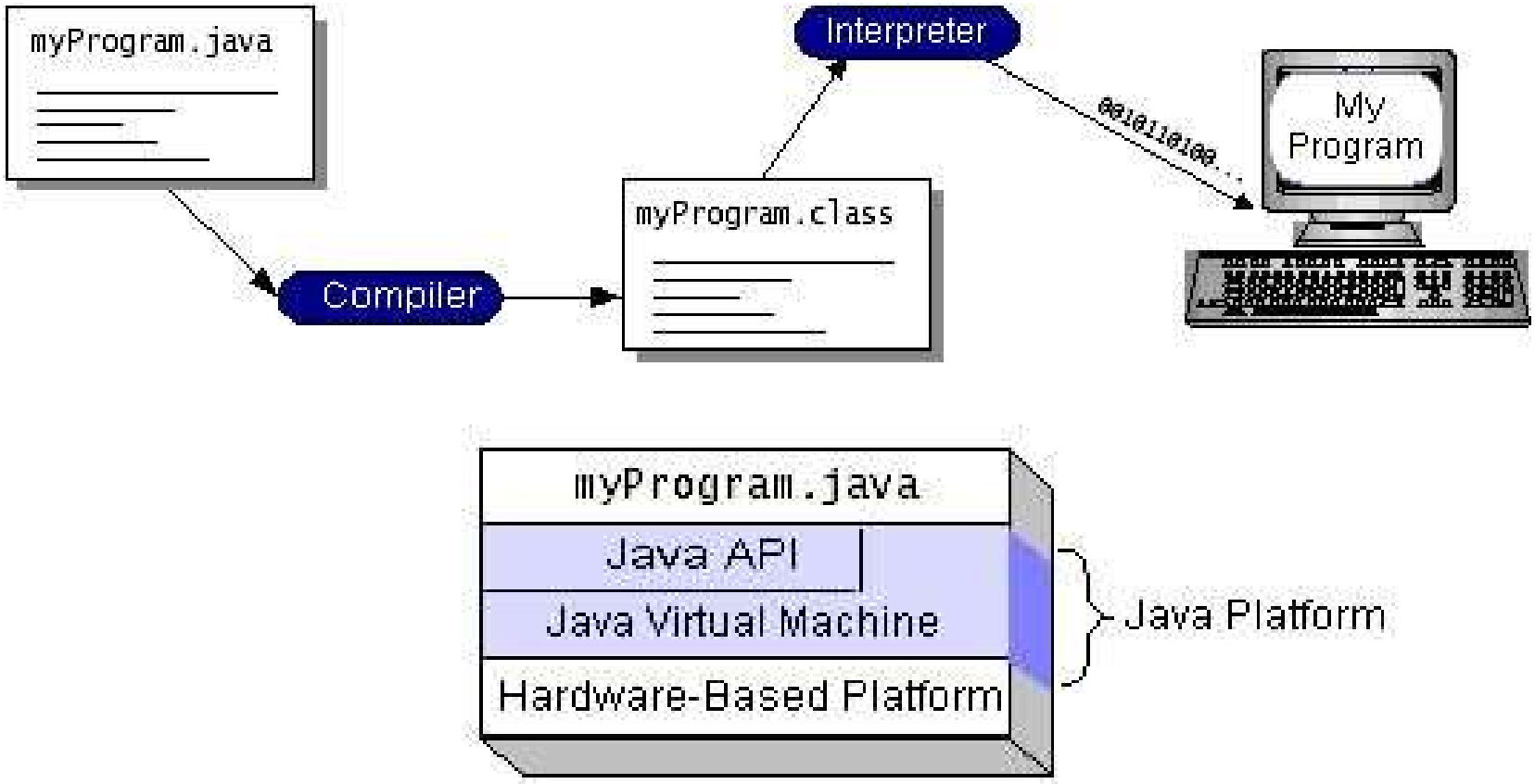
Interpretada, Neutra e Portável

- Bytecodes executam em qualquer máquina que possua uma JVM, permitindo que o código em Java possa ser escrito independente da plataforma
- A característica de ser neutra em relação à arquitetura permite uma grande portabilidade



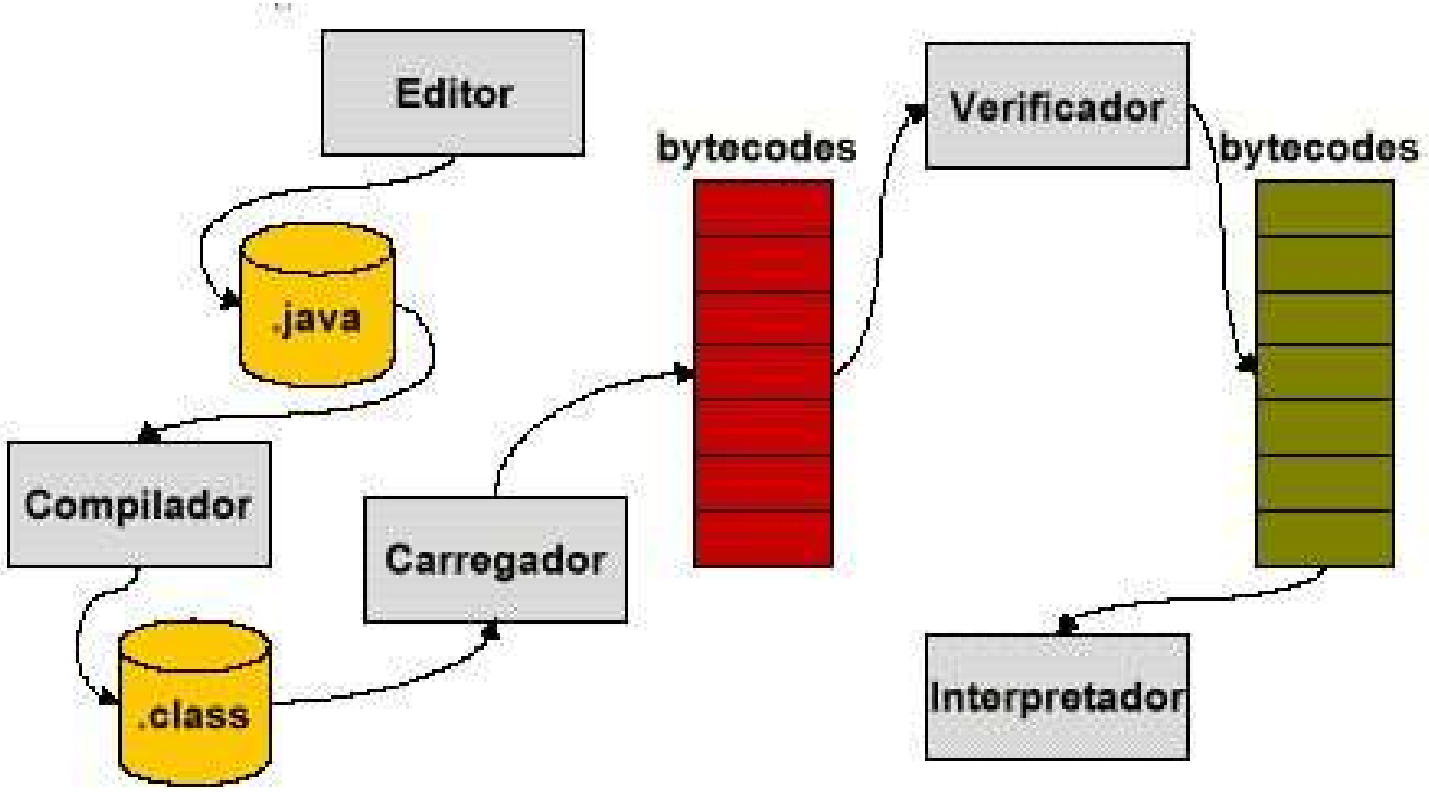
Fonte: MONTENEGRO, Anselmo. Introdução à Linguagem de programação Java - Aula de Programação de Computadores II. Niterói - RJ: Instituto de Computação UFF, s.a..

Interpretada, Neutra e Portável



Fonte: MONTENEGRO, Anselmo. Introdução à Linguagem de programação Java - Aula de Programação de Computadores II. Niterói - RJ: Instituto de Computação UFF, s.a..

O Ambiente Java



Fonte: MONTENEGRO, Anselmo. Introdução à Linguagem de programação Java - Aula de Programação de Computadores II. Niterói - RJ: Instituto de Computação UFF, s.a..

Ambiente de Desenvolvimento em Java

- Java possui um ambiente de desenvolvimento de software denominado Java SDK (Software Development Kit – antigamente denominado JDK)
    - É necessário instalar o kit para desenvolvimento de software Java, ou JDK (Java Development Kit)
- Não é um ambiente integrado de desenvolvimento, não oferecendo editores ou ambiente de programação
- O Java SDK contém um amplo conjunto de APIs (Application Programing Interface)

Ambiente de Desenvolvimento em Java

- Algumas ferramentas do Java SDK:
  - Compilador Java (javac)
    - Gera bytecodes a partir de código-fonte
  - Interpretador de aplicações Java (java)
    - interpreta (ou compila, se suportar JIT) os bytecodes para linguagem de máquina
  - Interpretador de applets Java (appletsviewer)

Ambiente de Desenvolvimento em Java

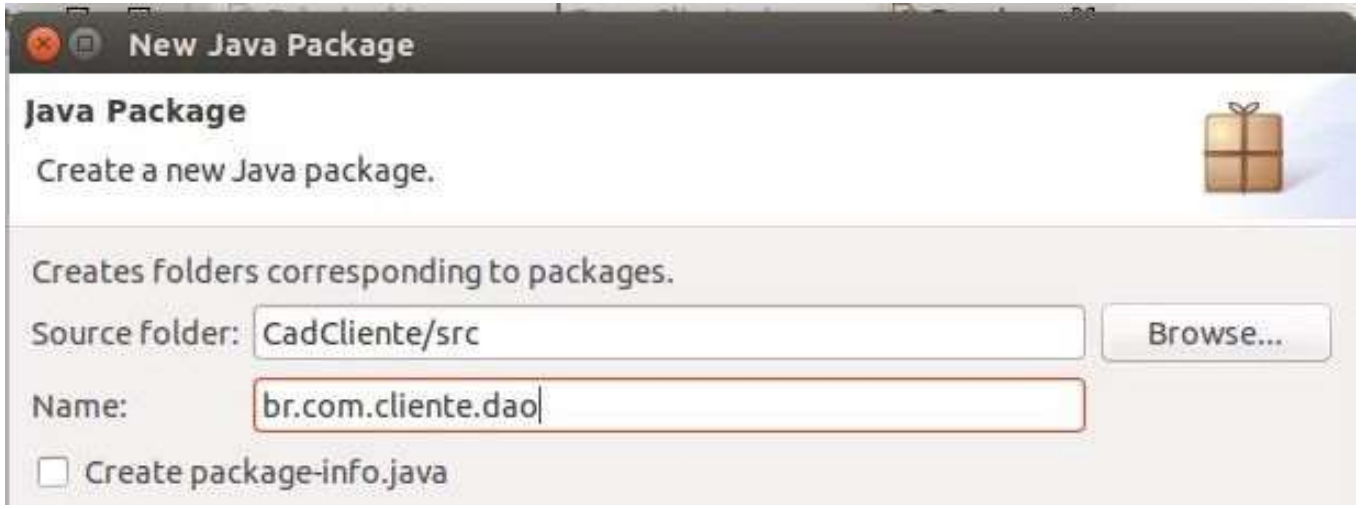
- Ainda existe o:
  - javadoc (um gerador de documentação para programas Java)
  - jar (o manipulador de arquivos comprimidos no formato Java Archive)
  - jdb (um depurador de programas Java), entre outras ferramentas

Estrutura de um programa em Java

- Um programa é composto por uma ou mais classes
- Tipicamente, cada classe é escrita em um arquivo fonte separado, cujo nome deve ser o mesmo da classe, com o sufixo .java
    - Exemplo: a classe “Pilha” deverá estar armazenada no arquivo Pilha.java
- Em geral, todas as classes que compõem um programa deverão estar no mesmo diretório

Packages

- Um pacote ou package na tecnologia Java nada mais é do que um conjunto de classes localizadas na mesma estrutura hierárquica de diretórios
- Com o uso de pacotes podemos organizar de forma física algo lógico (um grupo de classes em comum) que serão armazenados fisicamente em uma pasta



Packages

- Para indicar que as definições de um arquivo fonte Java fazem parte de um determinado pacote, a primeira linha de código deve ser a declaração de pacote:
  - `package nome_do_pacote`
- Caso tal declaração não esteja presente, as classes farão parte do "pacote default", que está mapeado para o diretório corrente

Packages

- Referenciando uma classe de um pacote no código fonte:
    - import nome_do_pacote.xyz ou simplesmente
    - import nome_do_pacote*
- Com isso a classe Xyz pode ser referenciada sem o prefixo nome_do_pacote no restante do código
- A única exceção refere-se às classes do pacote java.lang

Por hoje é só, próxima aula continuamos com JAVA ;)