Instituto de computação

Técnicas de Programação Avançada

TCC-00175

Profs.: Anselmo Montenegro

www.ic.uff.br/~anselmo

Conteúdo: Introdução à Orientação a Objetos

Programação Estruturada

Composição dos Programas:

Um programa é composto por um conjunto de rotinas
A funcionalidade do programa é separada em rotinas
Os dados do programa são variáveis locais ou globais

Fluxo de Execução:

O programa tem início em uma rotina principal
A rotina principal chama outras rotinas
Estas rotinas podem chamar outras rotinas, sucessivamente
Ao fim de uma rotina, o programa retorna para a chamadora

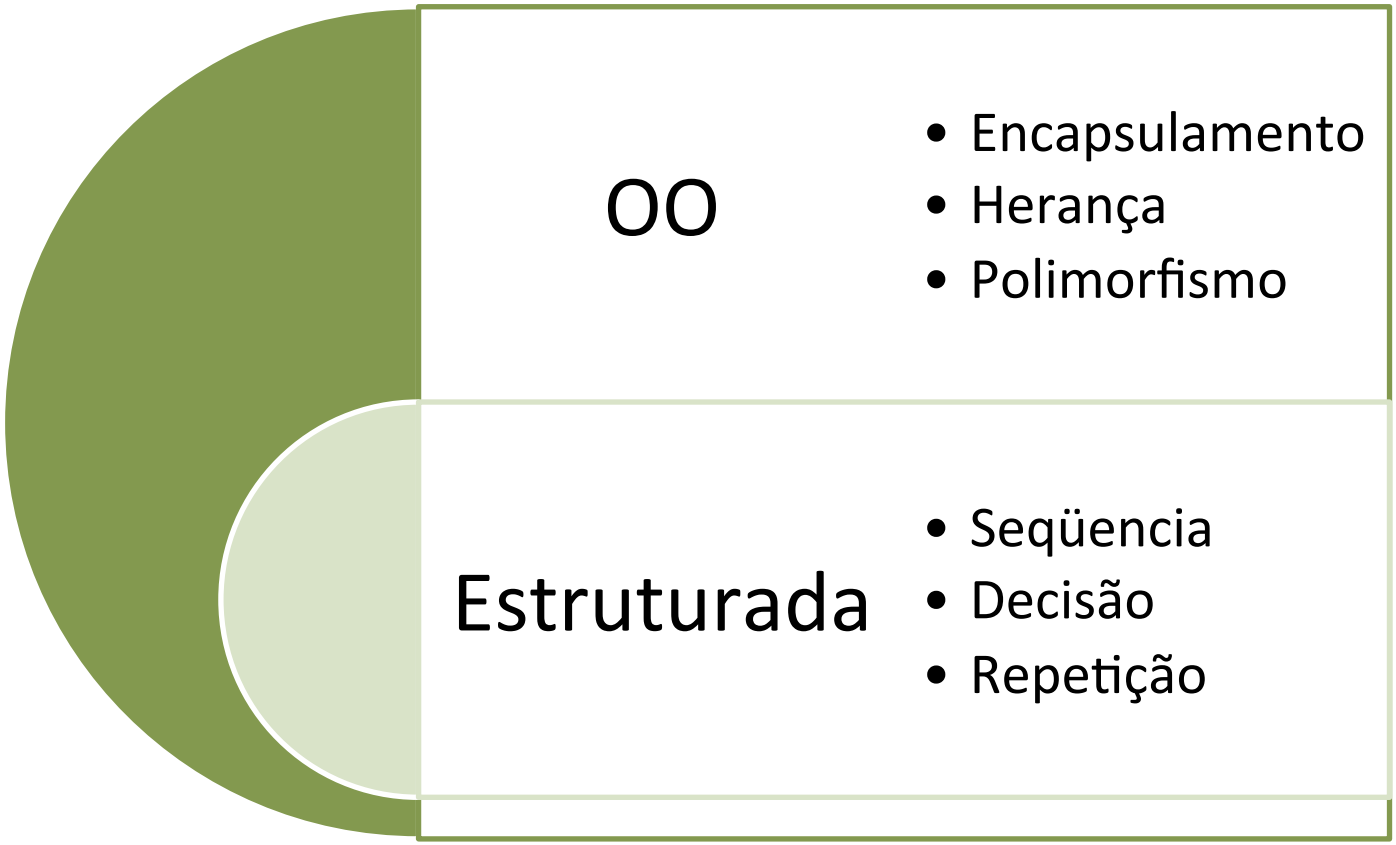
Programação Orientada a Objetos

Composição do programa:

A funcionalidade do programa é agrupada em objetos
Os dados do programa são agrupados em objetos
Os objetos agrupam dados e funções correlacionados

Fluxo de Execução:

Similar ao anterior
Os objetos colaboram entre si para a solução dos objetivos
A colaboração se realiza através de chamadas de rotinas


OO
• Encapsulamento
• Herança
• Polimorfismo
Estruturada
• Seqüencia
• Decisão
• Repetição

O ser humano se relaciona com o mundo através do conceito de objetos.

Estamos sempre identificando objetos ao nosso redor.

Para isso:

atribuímos nomes
classificamos em grupos – **classes**.

Definição

Um objeto é a representação computacional de um elemento ou processo do mundo real

Cada objeto possui um conjunto de características e comportamentos

Definição

Uma característica descreve uma propriedade de um objeto, ou seja, algum elemento que descreva o objeto

Exemplo de características de um objeto identificado como **carro:**

Cor
Marca
Número de portas
Ano de fabricação e tipo de combustível

Definição:

Um comportamento representa uma ação ou resposta de um objeto a uma ação do mundo real.

Exemplos de comportamento para o objeto **carro**

Acelerar
Parar
Andar
Estacionar

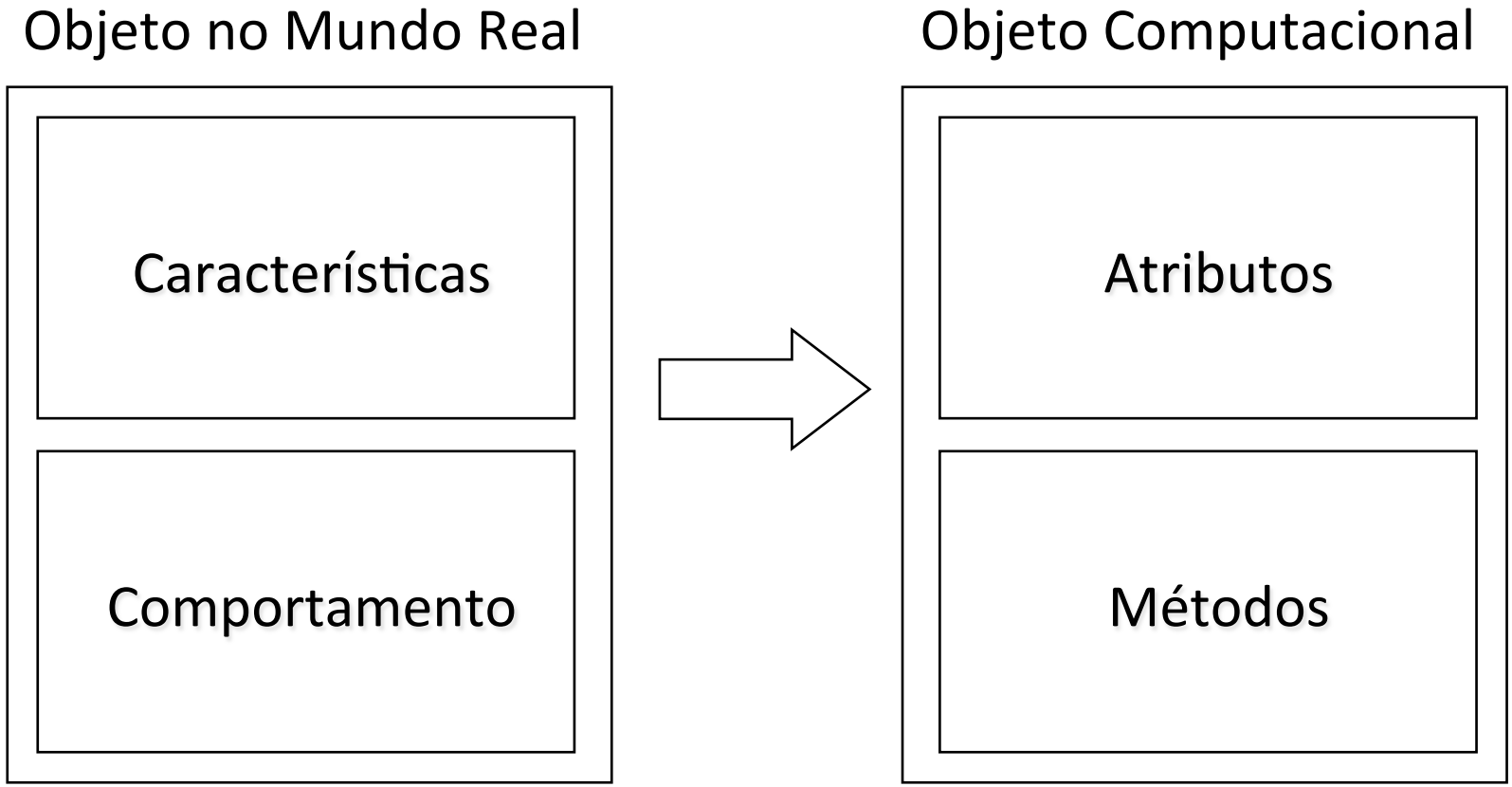
Exemplos:

- ✓ Cachorros
  características: nome, cor, raça
  comportamentos: latir, correr

- ✓ Bicicletas
  características: marcha atual, velocidade atual
  comportamentos: trocar marcha, aplicar freios

Orientação a Objetos

Mapeamento entre objetos reais e computacionais




Identidade
Classificação
Encapsulamento
Polimorfismo
Hereditariedade
Ligação Dinâmica

Identificar as características e o comportamento de objetos do mundo real é o primeiro passo da programação OO.

Observe um objeto e pergunte:

1. Quais os possíveis características deste objeto e quais estados elas assumem?
2. Quais comportamentos (ações) que ele pode executar?


Identidade
Classificação
Encapsulamento
Hereditariedade
Polimorfismo
Ligação Dinâmica

Objetos não são considerados isoladamente

Um processo natural é identificar características e comportamentos semelhantes entre objetos

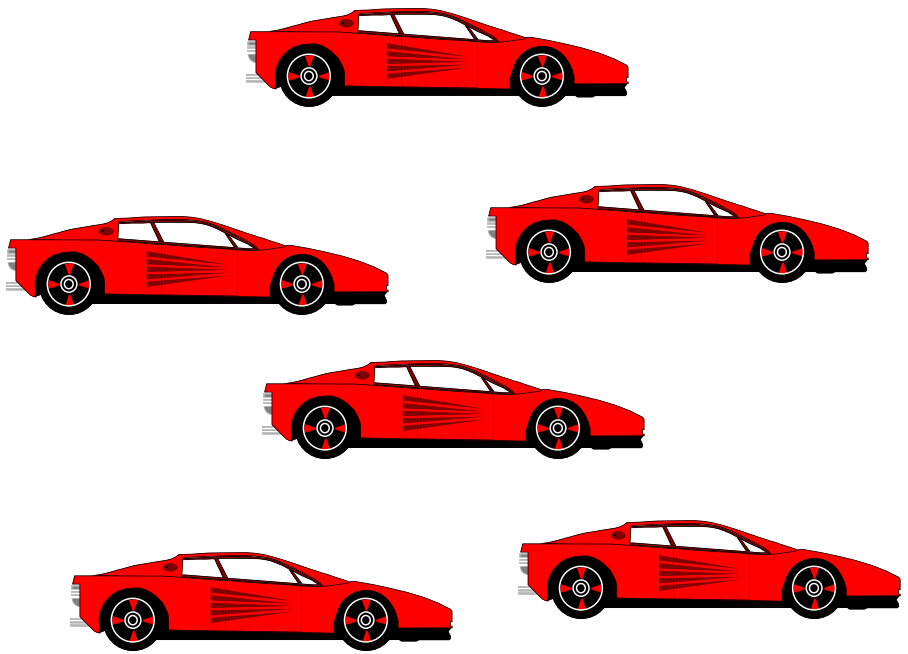
Objetos com características e comportamentos semelhantes são agrupados em classes

A unidade fundamental em programação em orientação a objetos (POO) é a **classe**.

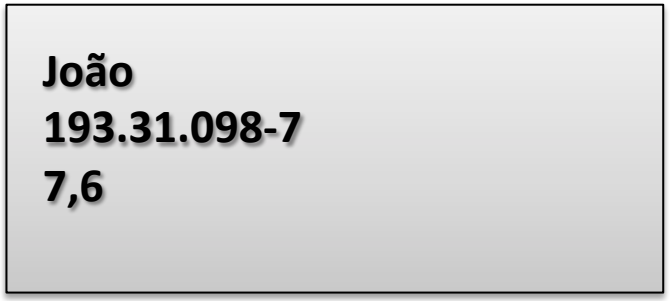
Classes contém:

- ✓ Atributos: determinam o estado do objeto;
- ✓ Métodos: semelhantes a procedimentos em linguagens convencionais, são utilizados para manipular os atributos.

Carro
Número de Rodas Cor Cor Lateral
Anda Para Acelera Estaciona

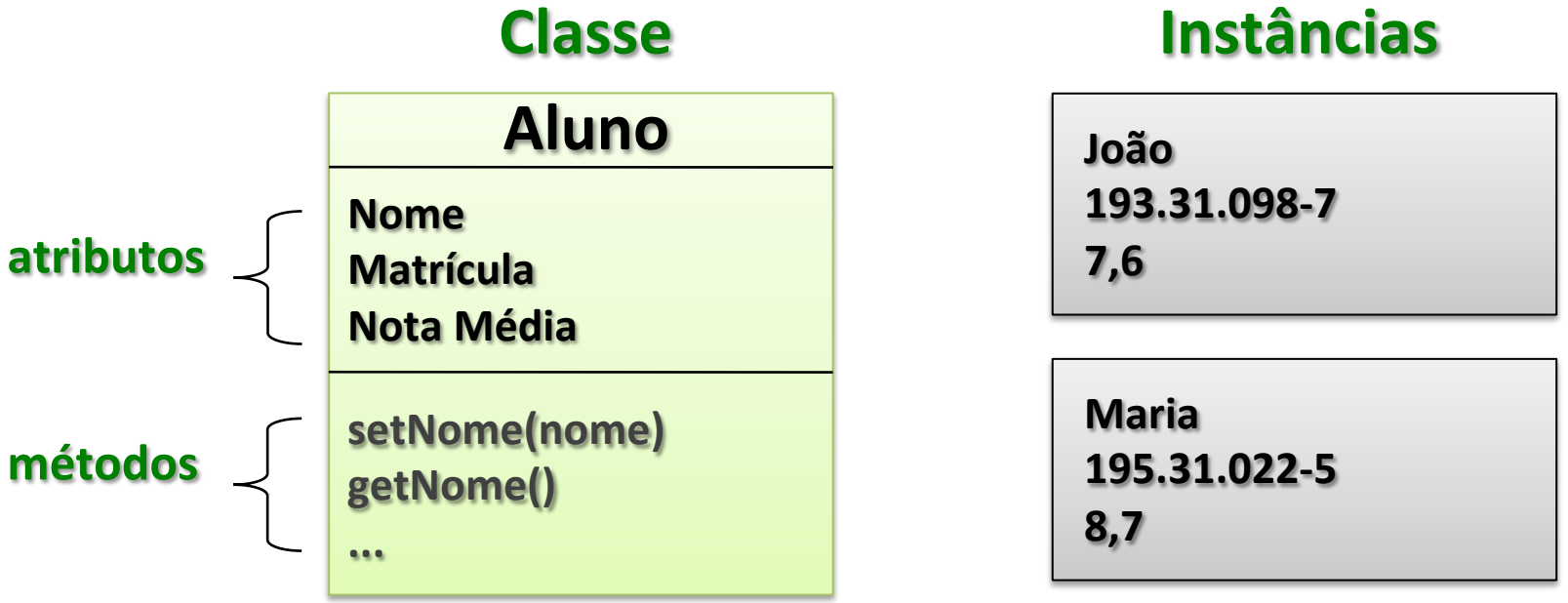


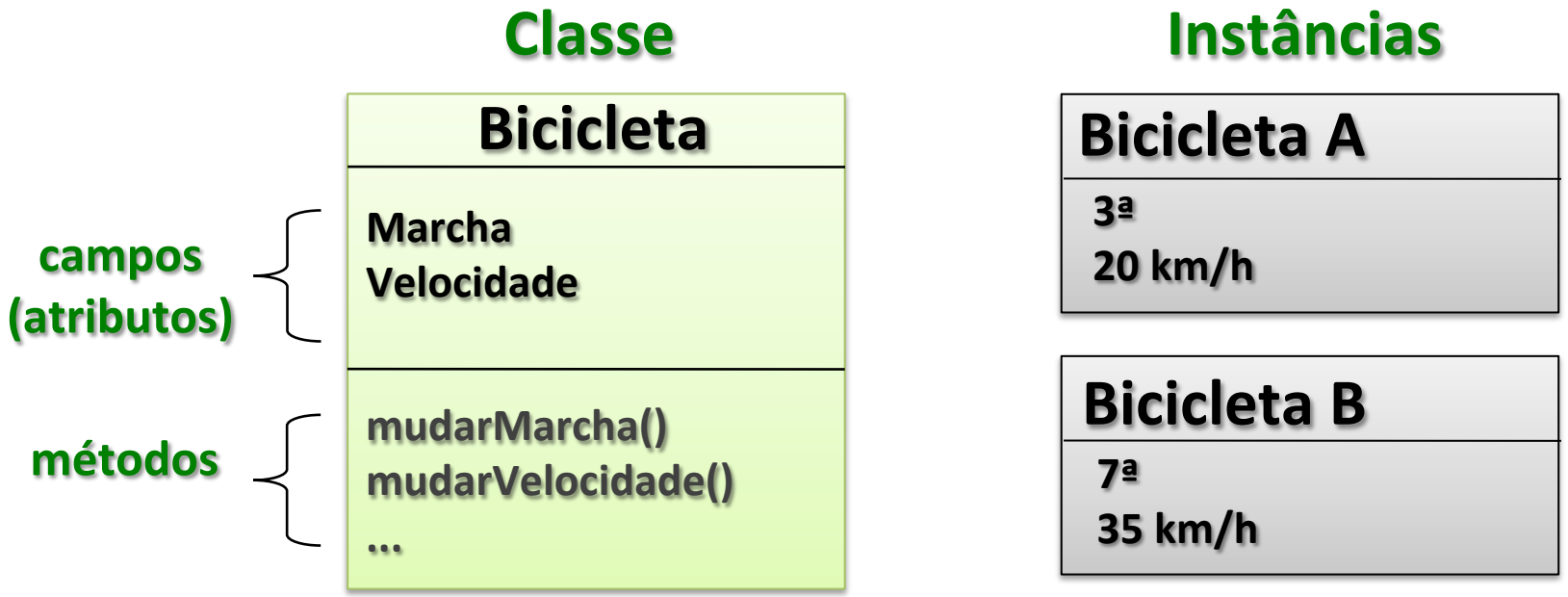

Aluno
Nome
Matrícula
Nota Média


João
193.31.098-7
7,6


Maria
195.31.022-5
8,7

As classes proveem a estrutura para a construção de objetos - estes são ditos **instâncias** das classes




Classe
Instâncias
Bicicleta
campos (atributos)
Marcha
Velocidade
métodos
mudarMarcha()
mudarVelocidade()
...
Bicicleta A
3ª
20 km/h
Bicicleta B
7ª
35 km/h

Um objeto é uma instância de uma única classe.

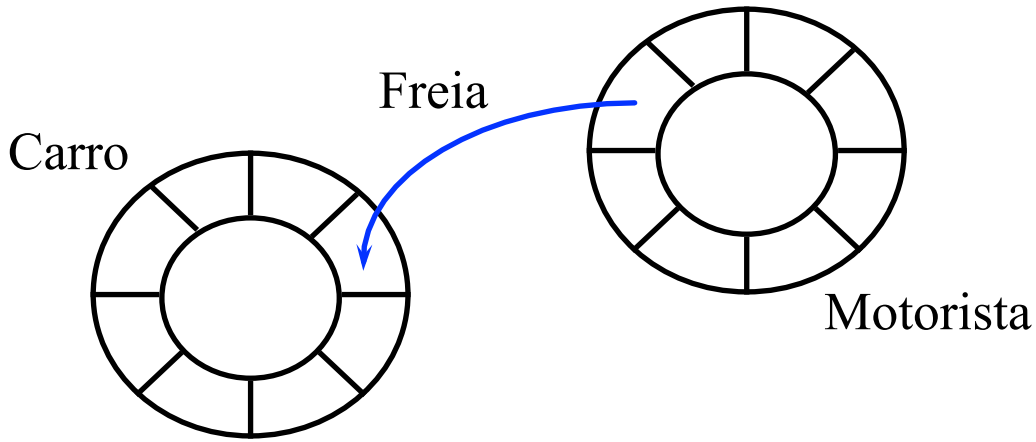
Uma instância de um objeto é uma unidade de programação que é armazenada em uma variável.

Um programa orientado a objetos é composto por um conjunto de objetos que interagem entre si.

Um programa OO é um conjunto de objetos que colaboram entre si para a solução de um problema

Objetos colaboram através de trocas de mensagens

A troca de mensagem representa a chamada de um método

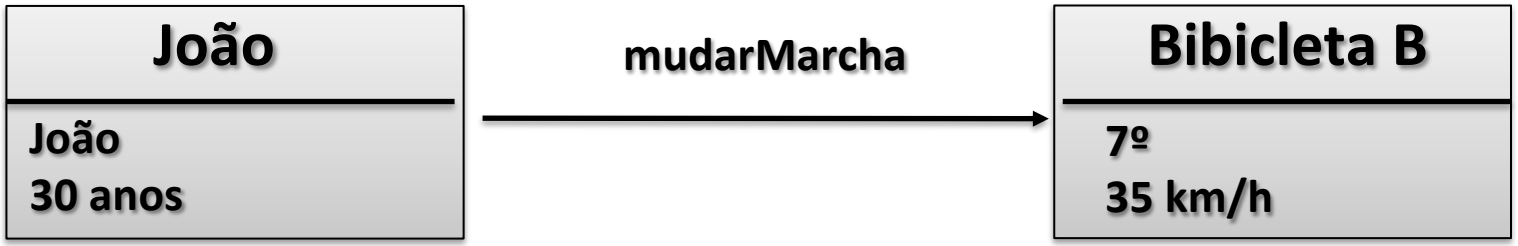


Um envio de mensagem sempre possui:

- Um emissor
- Um receptor
- Um seletor de mensagens (nome do método chamado)
- Parâmetros (opcionais)

Uma mensagem pode retornar um valor

Métodos operam no estado interno de um objeto e servem como mecanismo de comunicação entre objetos.



Orientação a Objetos

Classes em Java

```
Qualificador_de_acesso class Nome_Da_Classe
{
   // atributos da classe
   // métodos da classe
}
```

```
// Class Lampada
public class Lampada
{
     // Atributos
     boolean acesa;

     // Métodos
     public void ligar()
     {   acesa = true;   }
     public void desligar()
     {   acesa = false;  }
}
```

```
// Class Bicicleta
class Bicicleta
{
     // Atributos
     int velocidade = 0;
     int marcha = 1;

     // Métodos
     void mudarMarcha(int novoValor)
     {   marcha= novoValor;   }
     Void aumentaVelocidade(int incremento)
     {   velocidade+= incremento;  }
}
```

Para instanciarmos um novo objeto devemos utilizar o operador new, como nos exemplos abaixo:

```
NomeDaClasse  nomeDoObjeto = new NomeDaClasse();
```

- ✓ Criando dois objetos bicicleta:

```
Bicicleta bicicleta1 = new Bicicleta();
Bicicleta bicicleta2 = new Bicicleta();
```

- ✓ Invocando seus métodos:

```
bicicleta1.mudarMarcha(2);
bicicleta2.aumentaVelocidade(5);
```

A classe provê a estrutura para a construção de objetos.

Um objeto é uma instância de uma classe.

Contém um estado (valores de seus atributos).

Expõe o seu comportamento através de métodos (funções).


Identidade
Classificação
Encapsulamento
Hereditariedade
Polimorfismo
Ligação
Dinâmica

É um princípio fundamental da OO:

Esconder o estado interno (valores dos atributos).

Obrigar que interações com os atributos sejam executadas através de métodos.

Com o encapsulamento um objeto determina a permissão que outros objetos terão para acessar seus atributos (estado).

Definição

É a utilização de técnicas de programação e mecanismos de linguagem de programação para agrupar e restringir acesso à atributos métodos e classes

Objetivo:

Reduzir a complexidade externa (interface) das classes
Preservar a integridade dos dados internos dos objetos

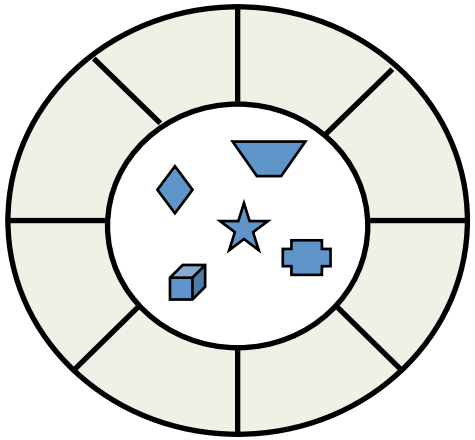
Encapsulamento de atributos: métodos de acesso

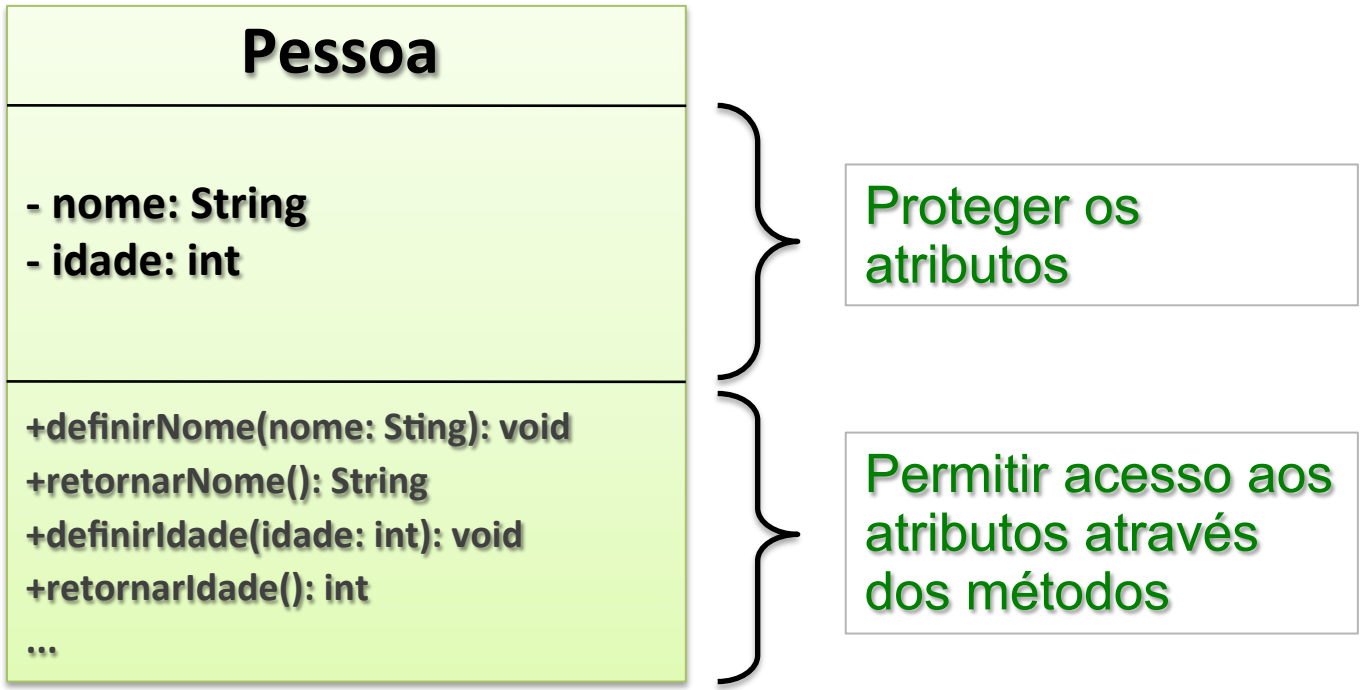
Encapsulamento de métodos: classes

Encapsulamento de classes: pacotes

Atributos e Métodos

- Os métodos formam uma "cerca" em torno dos atributos
- Os atributos não podem ser manipulados diretamente
- Os atributos somente podem ser alterados ou consultados através dos métodos do objeto




Pessoa
- nome: String
- idade: int
+definirNome(nome: Sting): void
+retornarNome(): String
+definirIdade(idade: int): void
+retornarIdade(): int
...
Proteger os atributos
Permitir acesso aos atributos através dos métodos

Um objeto X é denominado cliente de um objeto Y se utiliza métodos de Y

Pelo encapsulamento:

Clientes de um objeto podem utilizar seus métodos sem conhecer os detalhes de sua implementação

A implementação de um objeto pode ser alterada sem o conhecimento de seus clientes, desde que a interface visível seja mantida

Encapsulamento de Dados