

MODELAGEM DE SOFTWARE

Dr. Richard Henrique de
Souza

ORIENTAÇÃO A OBJETOS

- O **objeto** é uma **abstração** de conjunto de **coisas do mundo** real. Pode ser uma **máquina**, uma **organização**, um **carro**, uma **passagem de ônibus** ou **negócio**;
- A OO **modela objetos** do mundo real, estudando e criando **classes** a partir de suas **características** como cor, nome, tamanho, etc. (Lima, 2005).

ORIENTAÇÃO A OBJETOS

Domínio da Aplicação – modelo obtido dentro de um contexto de negócio após o estudo e observação da realidade.

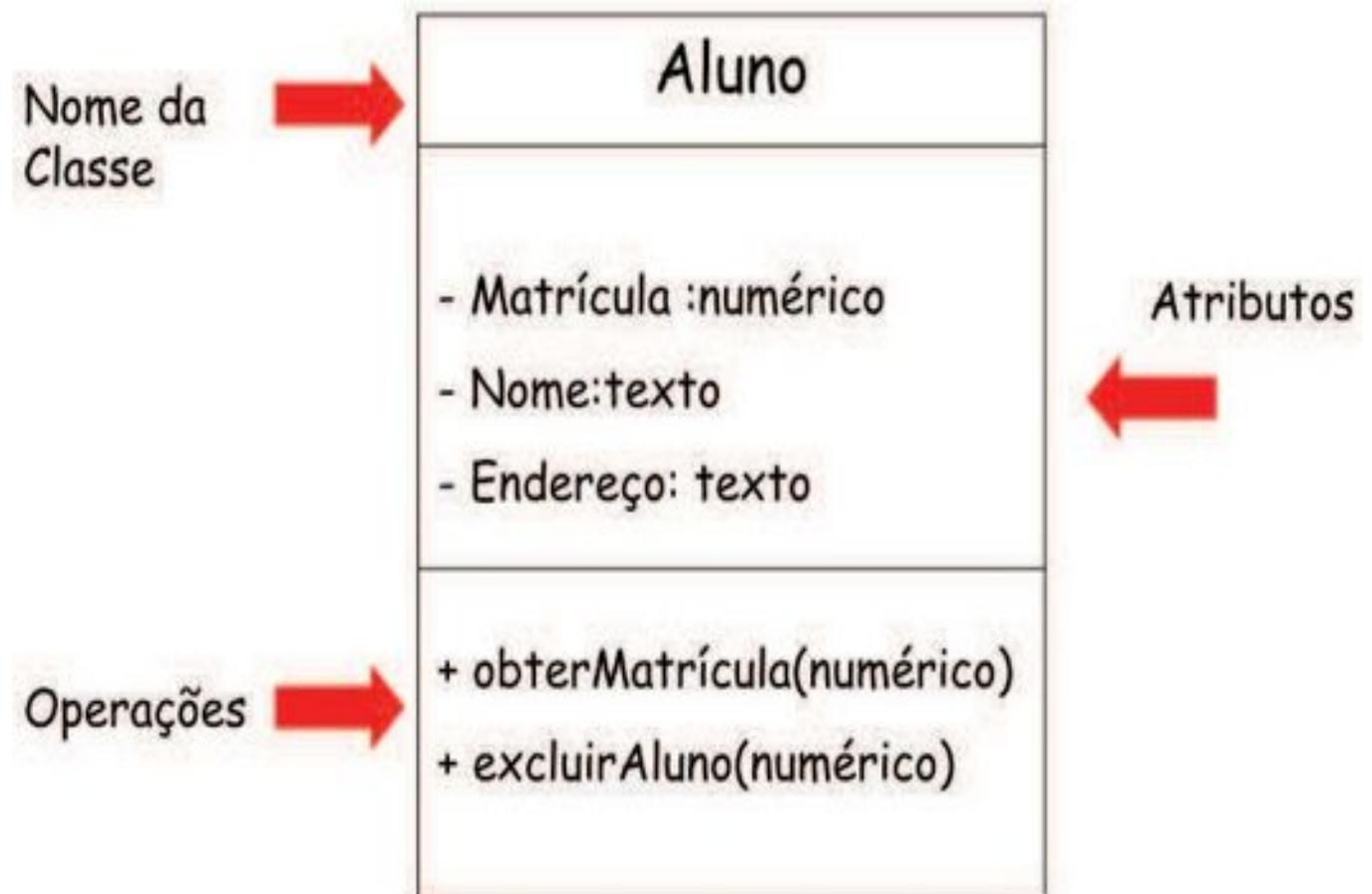
Abstração – processo de separar observados e mentalmente os objetos estudados da realidade.

ORIENTAÇÃO A OBJETOS

- **Classe** – pode ser vista como a **descrição de um tipo de objeto**, com propriedades semelhantes (atributos), o mesmo comportamento (operações), os mesmos relacionamentos com outros objetos e a mesma semântica.
- A classe agrupa objetos com características e comportamentos comuns!

ORIENTAÇÃO A OBJETOS

- Por exemplo:
 - a classe aluno de uma escola, apresenta um conjunto de alunos que apresentam as mesmas informações.



ORIENTAÇÃO A OBJETOS

- **Instância:**

- A classe por sua vez deve descrever as propriedades e comportamentos daquele objeto. Uma classe descreve um grupo de objetos;
- Cada objeto do mundo real pertencente a uma classe é denominado instância da classe;
- João Filomeno é então uma instância da classe Aluno.

ORIENTAÇÃO A OBJETOS

Atributos:

- O atributo é a descrição dos dados armazenados pelos objetos de uma classe;
- O atributo de uma classe está associado a um conjunto de valores que o atributo pode assumir;

ORIENTAÇÃO A OBJETOS

- São Atributos para a classe Aluno:
 - Matricula;
 - Nome;
 - Curso;
 - Idade;
 - Sexo...
- Os atributos **são sempre individuais** e cada objeto da classe possui seus próprios atributos.

ORIENTAÇÃO A OBJETOS

- **Operações:**

- As operações implementam serviços que podem ser solicitados por algum objeto da classe para modificar o comportamento.
- Todos os objetos da classe vão compartilhar destas operações.

VISIBILIDADE

- Público (+) – Qualquer classificador externo com visibilidade para que determinado classificador seja capaz de usar a característica.
- Protegido (#) – Qualquer descendente do classificador é capaz de usar a característica;
- Privado (–) – Somente o próprio classificador é capaz de usar a característica;
- Pacote (~) – Somente classificadores declarados no mesmo pacote podem usar a característica.

ENCAPSULAMENTO

- No paradigma OO, **o acesso aos dados dos objetos** deve ocorrer **somente pelos métodos dos próprios objetos** (membro de dados privados);
- Garantindo a **manipulação adequada** dos mesmos e utilização das regras de negócio implementadas;

ENCAPSULAMENTO

Também é conhecido com **ocultação de informação;**

Deste modo os objetos definem **quais serviços** estão **acessíveis** a outros **objetos;**

Um padrão proposto pelo mercado é associado a utilização de métodos de acesso:

- setters;
- getters.

MÉTODOS SETER

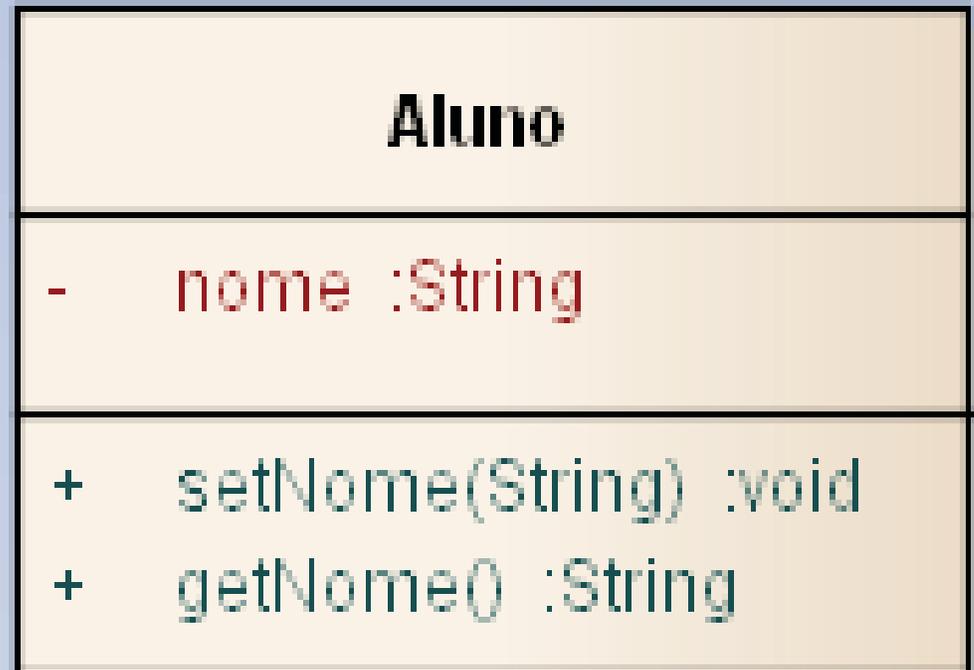
- Todos os métodos utilizados para alterar ou inserir valores nos membros de dados;
- No padrão proposto, esses métodos deve começar com a palavra set seguida do nome da propriedade.
- Por exemplos:
 - Atributo: `int idade;`
 - Método: `setIdade(int);`

MÉTODOS GETER

- Todos os métodos utilizados para recuperar valores encapsulados nos objetos;
- No padrão proposto, esses métodos devem começar com a palavra *get*, seguida pelo nome da propriedade recuperada;
- Por exemplo:
 - Atributo: `int idade;`
 - Método: `getIdade(): int;`

EXEMPLO

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



VANTAGENS DE USO

- Garantir que as **regras de negócio** relacionadas aos dados **sejam realmente utilizadas**;
- Impedindo que alterações inapropriadas nos atributos sejam realizadas;
- Garantir o correto acesso às informações.

MÉTODOS CONSTRUTORES

- Os métodos construtores, como o próprio nome sugere, tem como finalidade a construção de objetos;
 - Pode-se pensar como um **contrato** para a **construção do objeto**;
- O método construtor pode ser de duas formas:
 - Com argumento; e
 - Sem argumento.

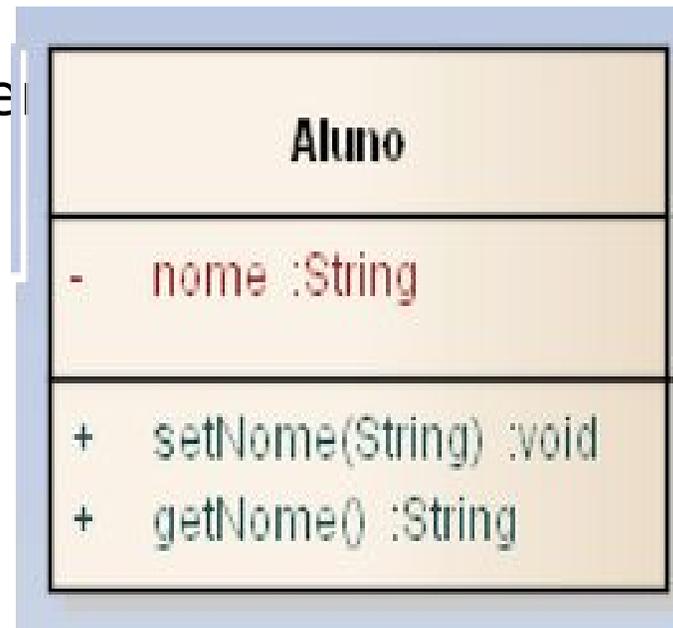
CONSTRUTOR SEM ARGUMENTO

Um construtor sem argumento sempre é criado em uma classe;

Ele inicia todos os atributos de classe com os seus valores default (nulo ou zero);

```
Aluno aluno = new Aluno();
```

nome se



CONSTRUTOR COM ARGUMENTO

- A possibilidade da **utilização** de um **método construtor com argumento**, traz um **segurança** para o desenvolvedor que concebe o objeto;
- Quando um objeto possui um construtor com argumento **ele está dizendo** a seguinte coisa para quem for manipular esse objeto:
 - Para utilizar esse objeto precisas preencher os seguintes atributos, atributos esses que estão sendo pedidos como parâmetros no construtor.

CONSTRUTOR ARGUMENTO

Criação do objeto Aluno:

```
Aluno aluno = new Aluno(98413);
```

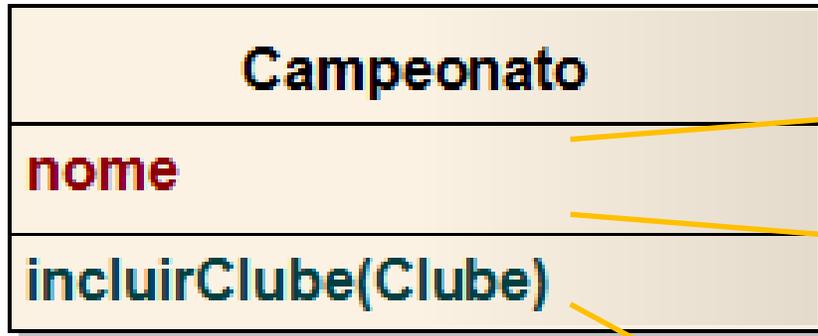
Aluno

- matricula :int
- nome :String

- + Aluno(int) :void
- + setMatricula(int) :void
- + getMatricula() :int
- + setNome(String) :void
- + getNome() :String

```
public class Aluno {  
  
    private int matricula;  
  
    private String nome;  
  
    public Aluno(int matricula) {  
        this.setMatricula(matricula);  
    }  
  
    public void setMatricula(int matricula) {  
        this.matricula;  
    }  
  
    public int getMatricula() {  
        return matricula;  
    }  
  
    public void setNome(String nome) {  
        this.nome = nome;  
    }  
  
    public String getNome() {  
        return nome;  
    }  
}
```

DIAGRAMA DE CLASSE - NOTAÇÃO



Nome da classe

Atributos da classe:

- atributos privados
- # atributos protegidos
- + atributos públicos

Métodos da classe

- métodos privados
- # métodos protegidos
- + métodos públicos

Atributos:

- **estáticos:** nome sublinhado
- **derivados:** /

Métodos:

- **estáticos:** nome sublinhado
- **abstratos:** nome em *itálico*

Classes:

- **abstratas:** nome em *itálico*

Tutoriais

<http://goo.gl/70Vys2>

<http://goo.gl/kjCNwb>

EXERCÍCIO: C01

TEMPO: 10 minutos



Faça um diagrama de classes indicando apenas os nomes das classes baseado nas *user stories* a seguir:

(QUESTAO adaptada do ENADE) Um analista de requisitos identificou as seguintes histórias de usuários para um sistema web de reserva de passagens de uma empresa aérea:

HST 01: COMO cliente, GOSTARIA de me cadastrar no site da empresa informando meu e-mail, nome, CPF, telefone e endereço PARA poder usar os serviços web da empresa.

HST 02: COMO cliente, GOSTARIA de pesquisar por preços das passagens aéreas informando data, hora, cidade e aeroporto de ida e de volta PARA poder escolher uma opção de reserva que me interessa.

HST 03: COMO cliente, GOSTARIA de confirmar uma reserva de passagem aérea selecionada, escolhendo uma das formas de pagamento disponibilizada pela empresa PARA poder viajar para o local escolhido na reserva.

HST 04: COMO cliente, GOSTARIA de fazer o check-in online da reserva PARA poder realizar meu embarque.

EXERCÍCIO: C02

TEMPO: 10 minutos



Agora adicione os atributos

Um analista de requisitos identificou as seguintes histórias de usuários para um sistema web de reserva de passagens de uma empresa aérea:

HST 01: COMO cliente, GOSTARIA de me cadastrar no site da empresa informando meu e-mail, nome, CPF, telefone e endereço PARA poder usar os serviços web da empresa.

HST 02: COMO cliente, GOSTARIA de pesquisar por preços das passagens aéreas informando data, hora, cidade e aeroporto de ida e de volta PARA poder escolher uma opção de reserva que me interessa.

HST 03: COMO cliente, GOSTARIA de confirmar uma reserva de passagem aérea selecionada, escolhendo uma das formas de pagamento disponibilizada pela empresa PARA poder viajar para o local escolhido na reserva.

HST 04: COMO cliente, GOSTARIA de fazer o *check-in* online da reserva PARA poder realizar meu embarque.

EXERCÍCIO: C03

TEMPO: 10 minutos



Agora adicione as operações

Um analista de requisitos identificou as seguintes histórias de usuários para um sistema web de reserva de passagens de uma empresa aérea:

HST 01: COMO cliente, GOSTARIA de me cadastrar no site da empresa informando meu e-mail, nome, CPF, telefone e endereço PARA poder usar os serviços web da empresa.

HST 02: COMO cliente, GOSTARIA de pesquisar por preços das passagens aéreas informando data, hora, cidade e aeroporto de ida e de volta PARA poder escolher uma opção de reserva que me interessa.

HST 03: COMO cliente, GOSTARIA de confirmar uma reserva de passagem aérea selecionada, escolhendo uma das formas de pagamento disponibilizada pela empresa PARA poder viajar para o local escolhido na reserva.

HST 04: COMO cliente, GOSTARIA de fazer o *check-in* online da reserva PARA poder realizar meu embarque.

DICAS - CURSOS

Fundamentos de Lógica de Programação
Introdução à Programação Orientada a Objetos (POO)
Linguagem de Modelagem Unificada (UML)
Ética no Desenvolvimento de Sistemas
Projetos de Sistemas de TI

- Fundação Bradesco / escola virtual
 - <https://www.ev.org.br/areas-de-interesse>

link de cursos de programação:

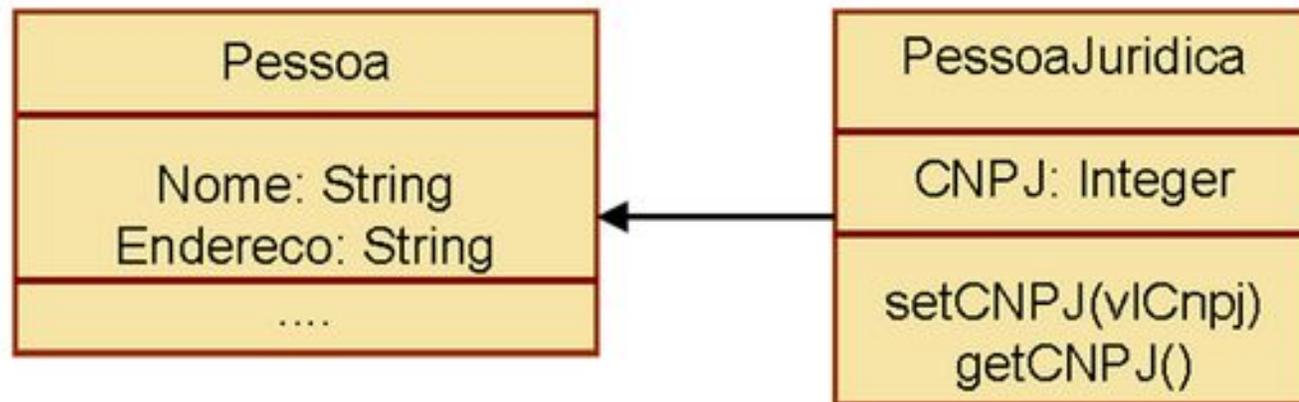
- <https://www.ev.org.br/areas-de-interesse/programacao>

Trilha: Fundamentos do Desenvolvimento de Sistemas

- <https://www.ev.org.br/trilhas-de-conhecimento/fundamentos-do-desenvolvimento-de-sistemas>

HERANÇA

É o dispositivo pelo qual objetos de uma classe D podem usar as operações ou atributos que de outro modo estaria disponível somente para os objetos da classe C, como se aquele métodos ou variável tivesse sido definido através de D.



HERANÇA

- É um relacionamento de especialização e generalização (“é um” ou “tipo de”).
- Herança define uma hierarquia de abstrações na qual uma subclasse herda de uma ou mais superclasses:
 - Herança simples: a subclasse herda de uma única superclasse.
 - Herança múltipla: a subclasse herda de duas ou mais Superclasses.

TEMPO: 15 minutos



EXERCÍCIO

Faça o diagrama de classes para a criação de uma conta no banco, considere que podemos ter contas correntes e contas poupança. A conta corrente possui um limite extra para saque. Já a conta poupança tem o rendimento mensal no aniversário da conta.

REFORÇANDO A BUSCA ATIVA

Leitura das páginas 114 até a página 125 e das páginas 141 até 144 do livro

Pressman, Roger, S. e Bruce R. Maxim. *Engenharia de software*. Disponível em: Minha Biblioteca, (9th edição). Grupo A, 2021.

Leitura da página 97 até a página 141 do livro

Horstmann, Cay. *Conceitos de Computação com Java*. Disponível em: Minha Biblioteca, Grupo A, 2009. (Inclusive é interessante realizar os exercícios de programação)

