

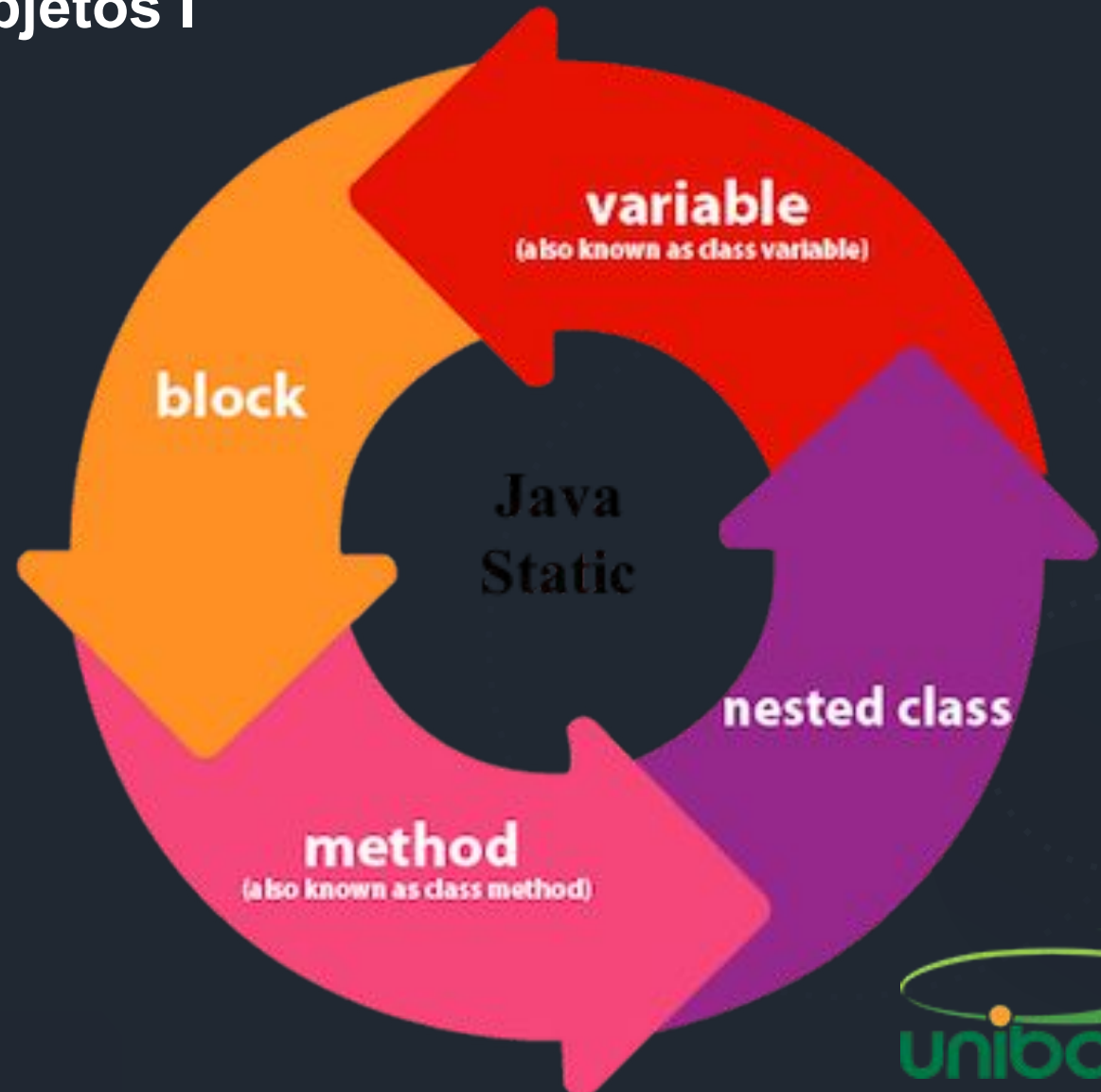
SISTEMAS DE INFORMAÇÃO

DISCIPLINA: Prog. Orientada a Objetos I

Professor: Clávison M. Zapelini

P00

Atributos e métodos
estáticos



ATRIBUTOS E MÉTODOS DE CLASSE (ESTÁTICOS)

Vimos até agora que os atributos e métodos pertencem ao objeto, ou seja, não se faz nada sem antes criar um objeto (new).

Há situações que é necessário utilizar atributos ou métodos de uma classe sem ter que criar um objeto.

Usando a palavra reservada **static** se define um atributo ou método de classe (estático) que pertence a classe como um todo e não ao objeto.

Pode-se acessar sem ter criado um objeto, utilizando diretamente o nome da classe. Objetos também podem acessá-los como membro do objeto, porém o valor continua compartilhado entre todos os objetos.

Métodos static não podem acessar atributos ou métodos não-static diretamente (precisa criar objeto).

ATRIBUTOS DE CLASSE (ESTÁTICOS)

```
public class Teste {  
    static int i = 47;  
    int j = 26 ;  
    public static void main(String[] args) {  
        Teste t1 = new Teste();  
        Teste t2 = new Teste();  
        System.out.println(t1.i + " - " + t1.j);  
        System.out.println(t2.i + " - " + t2.j);  
        t1.i++;  
        t1.j++;  
        System.out.println(t1.i + " - " + t1.j);  
        System.out.println(t2.i + " - " + t2.j);  
        Teste.i = 100;  
        System.out.println(t1.i);  
        System.out.println(t2.i);  
    }  
}
```

47 - 26

47 - 26

48 - 27

48 - 26

100

100

MÉTODOS DE CLASSE (ESTÁTICOS)

```
public class Teste {  
    static int i = 47;  
    int j = 26 ;  
  
    static void imprime(String s){  
        System.out.println(s);  
    }  
    static void incrementaI(){ i++;}  
    void incrementaJ(){ j++;}  
    public static void main(String[] args) {  
        Teste t1 = new Teste();  
        incrementaI();  
        Teste.incrementaI();  
        t1.incrementaI();  
  
        incrementaJ(); //ERRO  
        Teste.incrementaJ(); //ERRO  
        t1.incrementaJ();  
        imprime(t1.i + " - " + t1.j);  
    }  
}
```

ENCAPSULAMENTO: Técnica de programação OO que esconde a implementação do objeto, permitindo ao mesmo utilizar apenas a interface definida.

GET: Acesso padrão a atributo.

SET: Modificador padrão de atributo;

*Regra de nomenclatura: get / set + nome do atributo com a inicial em maiúscula:

Project Lombok

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



<https://dicasdejava.com.br/como-configurar-o-lombok-no-eclipse/>



ESPECIFICADORES DE ACESSO

Determinam a visibilidade de um determinado membro de classe (método ou atributo) com relação a outras classes.

Há quatro níveis de acesso:

Público (**public**)

Privado (**private**)

Protegido (**protected**)

Amigo (**friendly**)

Devem ser utilizadas antes do nome do membro a ser especificado.

Não podem ser utilizadas em conjunto.

ESPECIFICADORES DE ACESSO

MEMBROS PÚBLICOS

CLASSES	Classes públicas podem ser importadas em qualquer outra classe.
ATRIBUTOS	Atributos públicos podem ser lidos e alterados por qualquer outra classe.
MÉTODOS	Métodos públicos podem ser chamados por qualquer outra classe.

public

ESPECIFICADORES DE ACESSO

MEMBROS PRIVADOS

CLASSES	Somente classes internas* podem ser declaradas como privadas
ATRIBUTOS	Atributos privados somente podem ser lidos e alterados dentro da própria classe.
MÉTODOS	Métodos privados só podem ser chamados dentro da própria classe.

PRIVATE

* Classe interna é a declaração de uma classe dentro de outra.



ESPECIFICADORES DE ACESSO

MEMBROS AMIGOS

CLASSES	Classes amigas somente podem ser importadas dentro do mesmo pacote.
ATRIBUTOS	Atributos amigos somente podem ser lidos e alterados por classes que estão no mesmo pacote.
MÉTODOS	Métodos amigos somente podem ser chamados por classes que estejam dentro do mesmo pacote.

friendly



ESPECIFICADORES DE ACESSO

MEMBROS PROTEGIDOS

CLASSES	Somente classes internas* podem ser declaradas como protegidas.
ATRIBUTOS	Atributos protegidos somente podem ser lidos e alterados por classes dentro do mesmo pacote ou por subclasses**.
MÉTODOS	Métodos protegidos só podem ser chamados por classes dentro do mesmo pacote ou por subclasses

PROTECTED

* Classe interna é a declaração de uma classe dentro de outra.

** Conceito de herança (será visto nas próximas aulas)



ESPECIFICADORES DE ACESSO

	Classe	Pacote	Subclasse	Todos
public	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
protected	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
default	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
private	<input checked="" type="checkbox"/>			

EXERCÍCIO



Um time de futebol possui o nome e uma lista de jogadores.

Um jogador de futebol possui nome, número da camisa e a quantidade de gols marcados no campeonato

Defina as classes **Jogador** e **Time**, com os métodos que se fizerem necessários para atender aos seguintes requisitos:

- Cadastrar jogadores e times
- Listar todos jogadores de um time
- Verificar artilheiro do campeonato.
- Verificar qual time fez mais gols no campeonato;

Desenvolver testes unitários para verificar se as funcionalidades estão corretas.



EXERCÍCIO



Faça um programa que permita cadastrar livros em memória (utilizar ArrayList). Cada livro possui título, preço e autores (no mínimo 1 e no máximo 4 autores). Cada autor tem nome, sexo e idade.

Crie as classes com os atributos e métodos necessários para atender aos seguintes requisitos:

Cadastrar Autores e Livros, listar todos os livros cadastrados (todos os dados do livro, inclusive os autores com todos os dados), pesquisar por autor (listar todos os livros de um autor), pesquisar por faixa de valor do livro (mínimo e máximo), listar todos os livros cujo autores tenham crianças (idade ≤ 12), listar todos os livros que foram escritos apenas por mulheres ou por homens (o usuário informa qual sexo deseja realizar a consulta).

continua...



EXERCÍCIO

... continuação.



REGRAS:

O título do livro, deve ser exibido sempre em minúsculo (mesmo que o usuário faça o cadastro maiúsculo).

Não permitir cadastro de valores menores ou iguais a ZERO.

Ao cadastrar o autor, permitir apenas cadastro com nome e sobrenome (duas palavras).

Ao exibir o valor, mostrar o símbolo da moeda na frente (R\$)

A idade do autor não pode ser negativa ou zero

O sexo obrigatoriamente deve ser Masculino ou Feminino. (Enum)

Desenvolver testes unitários para verificar se as funcionalidades estão corretas.



EXERCÍCIO



Crie uma classe chamada "Funcionario" com os seguintes atributos estáticos:

- Um atributo inteiro chamado "totalFuncionarios" que armazena o número total de funcionários da empresa.
- Um atributo inteiro chamado "idadeMinima" que representa a idade mínima permitida para contratação.

A classe "Funcionario" deve ter os seguintes métodos estáticos:

- Um método chamado "getTotalFuncionarios" que retorna o número total de funcionários.

continua...



EXERCÍCIO



... **continuação.**

- Um método chamado "getIdadeMinima" que retorna a idade mínima permitida para contratação.
- Um método chamado "contratar" que recebe como parâmetros o nome e a idade de um funcionário, e verifica se a idade é maior ou igual à idade mínima permitida. Se for, o método deve incrementar o contador de total de funcionários e exibir uma mensagem de sucesso na contratação. Caso contrário, o método deve exibir uma mensagem informando que o funcionário não pode ser contratado devido à idade.

Requisitos: Exibir o número total de funcionários atualmente na empresa e contratar funcionários com nomes e idades diferentes.

Desenvolver testes unitários para verificar se as funcionalidades estão corretas.

