

# 03 - Codificação de classes, métodos e objetos

segunda-feira, 2 de maio de 2022 19:46

## ESTRUTURA DE UMA CLASSE

```
//declaração de pacotes
public class NomeDaClasse { //public pode ser omitido (será tratado em outro momento)

    //declaração de variáveis - variáveis de instância (atributos)
    //declaração dos métodos

}
```

atributos e métodos de uma classe são chamados de membros da classe.

```
public class Pessoa { // é uma classe pública

    String nome;
    int idade;

}
```

Atributos ou variáveis de instância (variáveis declaradas na classe)

OU

```
class Pessoa { // não é uma classe pública

    String nome;
    int idade;

}
```

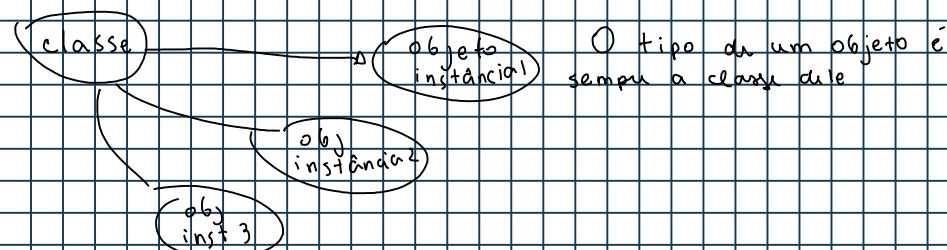
## CONVENÇÃO DE NOMEAÇÃO

classes	métodos	variáveis	constantes
- Camel Case - substantivos	- camelCase - verbo + substantivo	- camelCase	- MAIÚSCULO

## CONVENÇÃO DE NOMEAÇÃO DE ARQUIVO FONTE

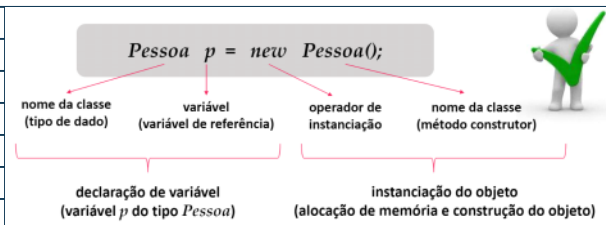
- apenas uma classe pública em cada arquivo-fonte
- se tiver classe pública, o nome do arquivo deve ser igual ao nome da classe
- um arquivo pode ter + de 1 classe não pública.

## INSTANCIACÃO DE CLASSES



sintaxe:

```
Classe classe = new Classe();
```



## ATRIBUTOS

também são conhecidos como dados membros e variáveis de instância.

```
public class Pessoa {
    String nome;
    int idade;
}
```

atributos ou variáveis de instância

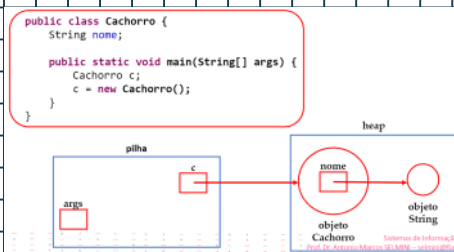
## MANIPULAÇÃO DE OBJETOS

`variávelDeReferência.atributo;`

operador ponto é utilizado para acessar atributos e métodos

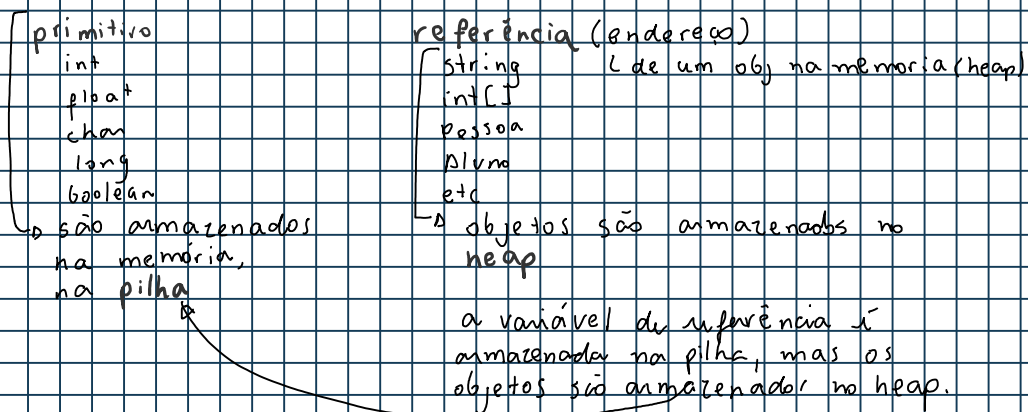
Prof. Dr. Antão

## PILHA E HEAP



## TIPO PRIMITIVO X REFERÊNCIA

Existem 2 tipos de dados em Java



## MÉTODOS

manipula o estado e descreve comportamentos de uma classe

# MÉTODOS

manipula o estado e descreve comportamentos da uma classe

assinatura do método

visibilidade tipo nomeDoMétodo(parâmetros) {  
corpo do método  
}

visibilidade: normalmente (**private**, **public**, **protected** ou **default**).

tipo: tipo retornado pelo método (por exemplo, **int**, **double**, etc.). Pode ser **void** para indicar que o método não tem retorno.

parâmetros: dados passados para o método.

Um método recebe parâmetros. Um chamador passa argumentos.

Métodos também retornam um tipo. Se o método não for retornar nada, usamos **void**.

visibilidade do método

saída do método

nome do método

valores de entrada do método (parâmetros)

assinatura do método

public int soma(int a, int b) {  
int r = a + b;  
return r;  
}

corpo do método