

Domain Driven Design - Java

Prof. Gilberto Alexandre das Neves
profgilberto.neves@fiap.com.br

Classes em Java

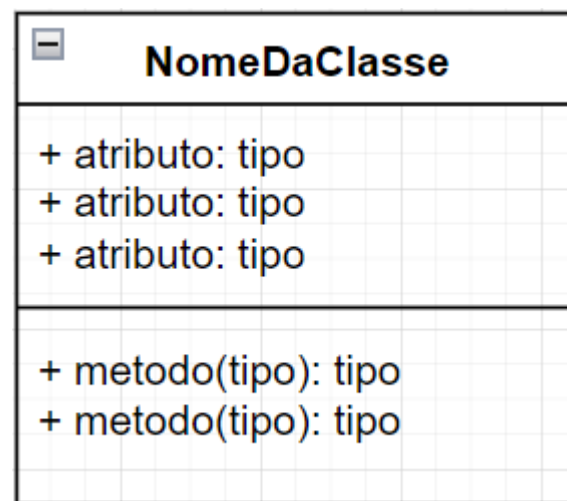
Uma **classe** é um molde, um modelo, um protótipo a partir do qual os objetos podem ser criados. Ao definir uma **classe**, podem ser criados muitos **objetos** a partir dela.

Uma classe é composta por seu **Nome**, seus **atributos** e **métodos**.

- Nomes de **Classes** inicia com letra maiúscula e a cada palavra nova a primeira letra também é maiúscula (não usar espaços ou caracteres especiais).
- Nomes de **atributos** e do **métodos** inicia com letra minúscula e a cada palavra nova a primeira letra é maiúscula (não usar espaços ou caracteres especiais).

Uma classe pode conter:

```
modificador class NomeDaClasse() {  
    // atributos  
    // métodos  
}
```



O **modificador** indica o modo de acesso por outras classes, define sua **visibilidade**.

Para **Classes**:

- **public** – a classe pode ser acessada por qualquer outra classe.
- *default* – a classe só pode ser acessada por classes no mesmo pacote. Este modificador é utilizado quando não se especifica um modificador.

Para **atributos e métodos**

- **public** – o código pode ser acessado por todas as classes.
- **private** – o código só pode ser acessado pela classe que a criou.
- *default* – o código somente é acessado no mesmo pacote. Este modificador é utilizado quando não se especifica um modificador.
- **protected** – o código somente é acessado no mesmo pacote e por **subclasses**.

Tipos de Atributos (variáveis)



Tipos de Variáveis

Tipos **primitivos** da linguagem, utilizados na criação de variáveis.

Tipo	Quantidade de bits	Valores
char	16	'\u0000' a '\uFFFF'
byte	8	-128 a + 127
int	32	-2.147.483.648 a +2.147.483.647
short	16	-32.768 a + 32.767
long	64	-9.223.372.036.854.775.808 a +9.223.372.036.854.775.807
float	32	-3.40292347E+38 a +3.40292347E+38
double	64	-1.79769313486231570E+308 a +1.79769313486231570E+308
boolean	8	true ou false

Tipos **não primitivos**:

Tipo	Quantidade de bits	Valores
String	??	cadeia de caracteres (usar aspas)

Operadores Aritméticos

Veja a seguir os operadores aritméticos usados na linguagem Java.

Função	Sinal	Exemplo
Adição	+	$x + y$
Subtração	-	$x - y$
Multiplicação	*	$x * y$
Divisão	/	x / y
Resto da divisão inteira	%	$x \% y$
Sinal negativo	-	$-x$
Sinal positivo	+	$+x$
Incremento unitário	++	$x++$
Decremento unitário	--	$x--$

Métodos

Métodos em Java

Um **método** é um bloco de código que só é executado quando é “chamado” (usado).

Você pode ter métodos onde se é possível passar dados (valores), chamados de parâmetros.

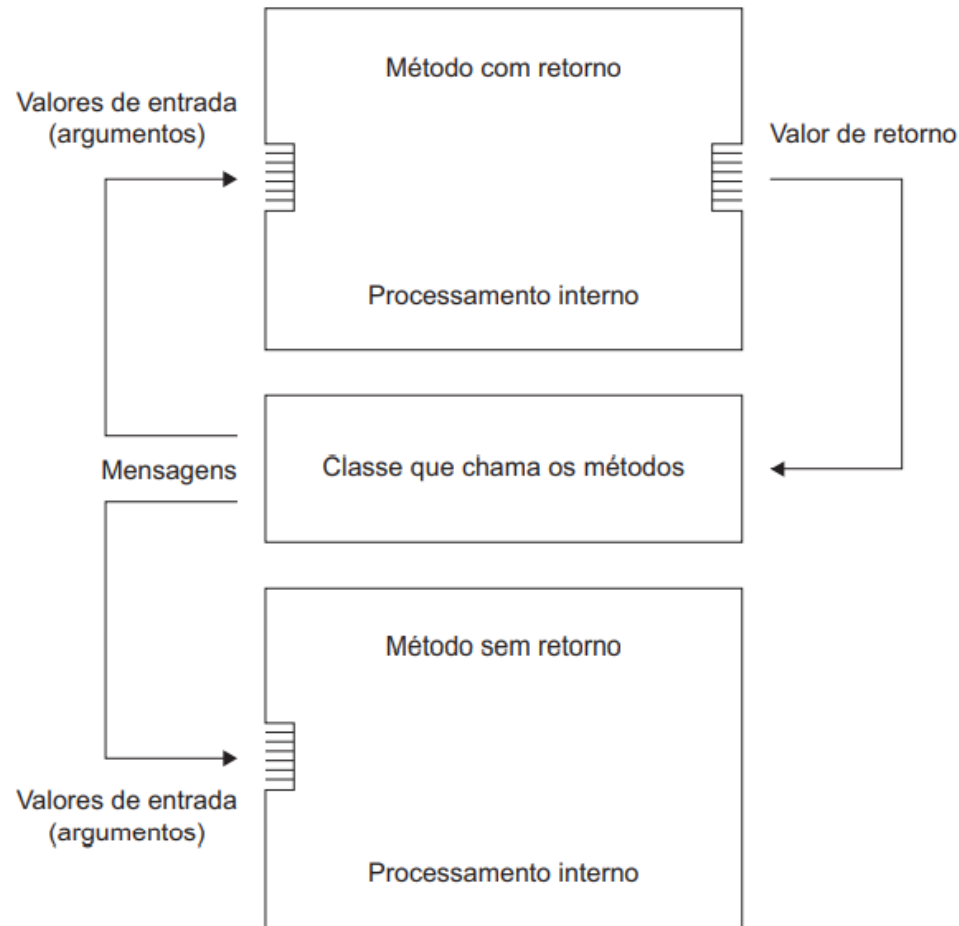
Em outras linguagens os métodos podem ser chamados de funções ou até mesmo procedimentos.

Porque usar métodos?

Para reutilizar o código (defina o código apenas uma vez e use-o várias vezes)

Métodos em Java

Podemos construir **métodos com retorno** ou **métodos sem retorno** (tudo vai depender da situação problema a ser resolvida). Veja a imagem abaixo para tentar entender a diferença entre eles:



| Sintaxe de um Método

Veja a seguir a sintaxe para criar um método (com retorno ou sem):

Método com retorno

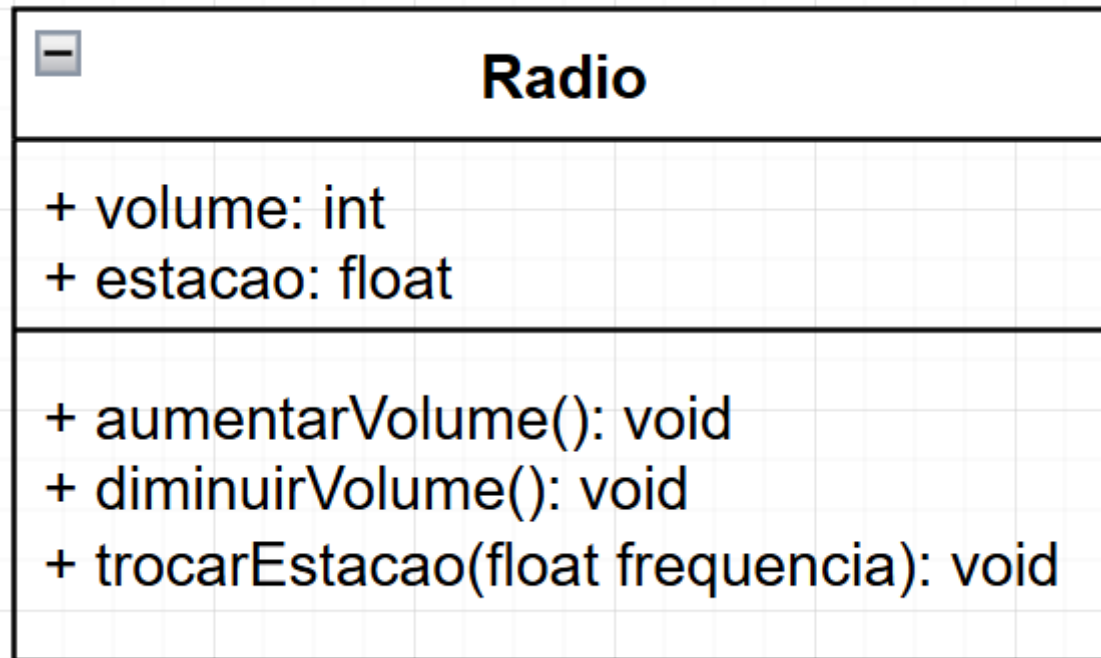
```
modificador tipo nomeDoMetodo(Lista-de-parâmetros) {  
    código do corpo  
    return valor-de-retorno;  
}
```

Método sem retorno

```
modificador void nomeDoMetodo(Lista-de-parâmetros) {  
    código do corpo  
}
```

A classe Radio

Veja a representação da classe **Radio** de acordo com a **UML** (Unified Modeling Language - uma linguagem de modelagem muito usada no mercado para representar certas características do software).



A classe Radio

Codifique a classe Radio como indicado abaixo:

```
1  package br.com.fiap;
2
3  public class Radio {
4      // atributos
5      public int volume;
6      public float estacao;
7
8      // métodos
9      public void aumentarVolume() {
10         volume++;
11     }
12     public void diminuirVolume() {
13         volume--;
14     }
15     public void trocarEstacao(float frequencia) {
16         estacao = frequencia;
17     }
18 }
```

Utilizando objetos da classe

Objetos da classe

Para utilizar um objeto de uma classe existem devemos efetuar:

- **A declaração do objeto:** segue o mesmo padrão de declaração das variáveis. Sua sintaxe é: `NomeDaClasse nomeDoObjeto;`
- **Instanciação do objeto:** corresponde à criação do objeto pela alocação de memória para armazenar informações sobre ele. Para realizar a instanciação do objeto utilizamos o operador **new** seguido do **nome da Classe** ().

Exemplo1:

```
Radio radio1;  
radio1 = new Radio();
```

Exemplo2:

```
Radio radio2 = new Radio();
```


A classe UsaRadio

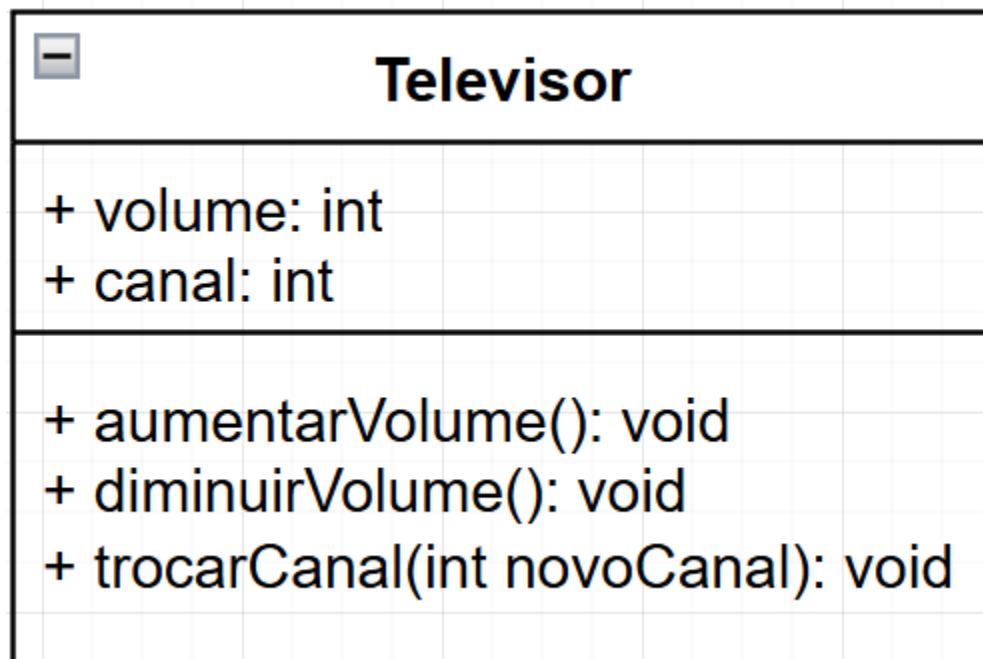
Implemente a classe abaixo com o método **main** para testar nossa classe **Radio**.

```
1 package br.com.fiap;
2
3 public class UsaRadio {
4     public static void main(String[] args) {
5         Radio radio1 = new Radio();
6         radio1.estacao = 89.1f;
7         radio1.volume = 5;
8         radio1.trocarEstacao(92.5f);
9         radio1.aumentarVolume();
10        radio1.aumentarVolume();
11        System.out.println("Volume atual: " + radio1.volume +
12                             "\nEstação atual: " + radio1.estacao);
13    }
14 }
```

Praticando...

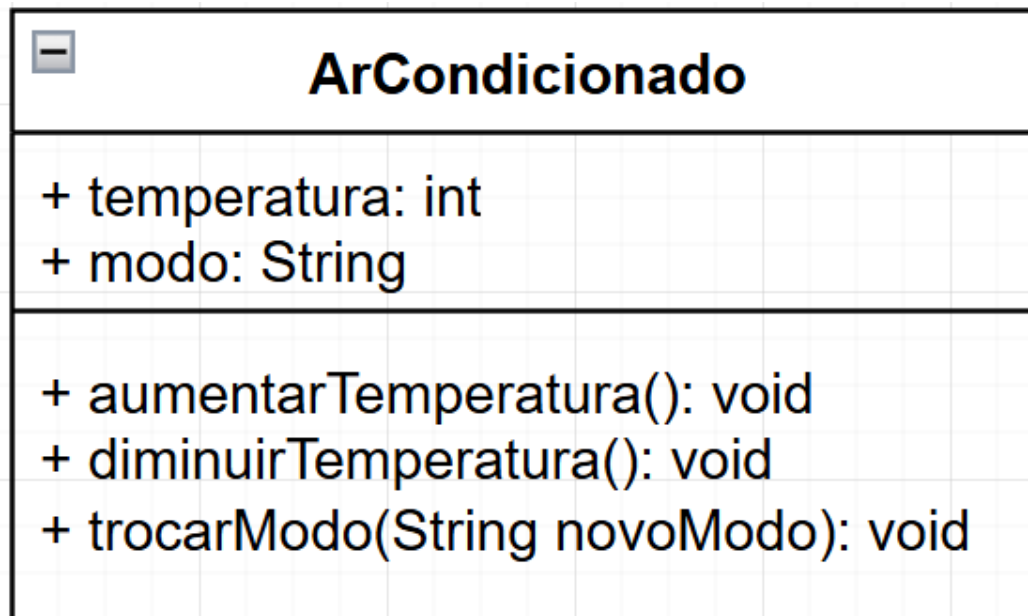
Praticando

Implemente a classe abaixo:



Agora crie uma nova classe (**UsaTelevisor**) com o método **main** para criar objetos e testar objetos da classe **Televisor**. Ao final exiba o volume e canal atuais.

Implemente a classe abaixo:



Agora crie uma nova classe (**UsaArCondicionado**) com o método **main** para criar objetos e testar a classe **ArCondicionado**. Ao final exiba a temperatura e o modo atuais.

Como valores para o atributo modo, utilize: “resfriar”, “aquecer” ou “ventilar”.

Use a criatividade e represente no modelo UML uma classe a sua escolha, defina para essa classe no mínimo 3 atributos e no mínimo 2 métodos.

Agora implemente o código desta classe no Java.

Em seguida, crie outra classe para criar um objeto da classe implementada anteriormente e modifique os valores de seus atributos, utilize seus métodos e exiba resultados.

1. Uma classe é composta por:
 - a) Nome, atributos e funções
 - b) Nome, procedimentos e métodos
 - c) Nome, atributos e métodos
 - d) Nome, comportamento e funções

2. Para instanciar um objeto de uma classe utilizamos o operador:
 - a) private
 - b) new
 - c) throw
 - d) public



Java como programar. Paul Deitel e Harvey Deitel. Pearson, 2011.

Java 8 – Ensino Didático : Desenvolvimento e Implementação de Aplicações. Sérgio Furgeri. Editora Érica, 2015.

Até breve!

Questionário - Resolução

1. Uma classe é composta por:
 - a) Nome, atributos e funções
 - b) Nome, procedimentos e métodos
 - ☒ c) Nome, atributos e métodos
 - d) Nome, comportamento e funções

2. Para instanciar um objeto de uma classe utilizamos o operador:
 - a) private
 - ☒ b) new
 - c) throw
 - d) public