

# ECM251 - Linguagens I

Teoria - Atributos de Classe  
Prof. Murilo Zanini de Carvalho  
Prof. Tiago Sanches da Silva

Antes de começar!

# Clone seu repositório do Github

- Lembre-se sempre antes de iniciar uma aula, clonar seu repositório remoto e realizar as atividades nele.
- Para cada atividade desenvolvida, criar um novo diretório.



# GitHub

Retirado de  
([https://miro.medium.com/max/4000/0\\*MZMI76wKo2FQLqG0.png](https://miro.medium.com/max/4000/0*MZMI76wKo2FQLqG0.png)), em 07/03/2021

# Atributos de Classes

# Atributos de Classes

Quanto precisamos compartilhar alguma informação entre todas as instâncias de uma classe, podemos criar um atributo compartilhado por todas elas.

Assim, quando um objeto modificar esse atributo, todos os outros acessar esse dado já modificado.

Para fazer isso em Java, declaramos o atributo como ***static***.

Quando um atributo é declarado como static, ele não pertence a cada objeto, mas sim a classe, deixando de ser individual para cada objeto.

Para acessarmos um atributo estático, não utilizamos a palavra this, mas sim o nome da classe.

# Atributos de Classes

Da mesma forma que os atributos estáticos, podemos utilizar método estáticos. Esses métodos podem ser acessados mesmo quando nenhuma instância da classe foi criada.

**ATENÇÃO:** métodos estáticos não podem acessar atributos não estáticos da classe.

# Atributos de Classes

## **Métodos e atributos estáticos**

Métodos e atributos estáticos só podem acessar outros métodos e atributos estáticos da mesma classe, o que faz todo sentido já que dentro de um método estático não temos acesso à referência `this`, pois um método estático é chamado através da classe, e não de um objeto.

O `static` realmente traz um "cheiro" procedural, porém em muitas vezes é necessário.

# Validador de CPF



# Validador de CPF

Criar uma classe que possa validar CPF informados para ela.

Mais informações:

<https://dicasdeprogramacao.com.br/algorithmo-para-validar-cpf/>

# Validador de CPF

Vamos usar como exemplo, um CPF fictício "529.982.247-25".

## Validação do primeiro dígito

Primeiramente multiplica-se os 9 primeiros dígitos pela sequência decrescente de números de 10 à 2 e soma os resultados. Assim:

$$5 * 10 + 2 * 9 + 9 * 8 + 9 * 7 + 8 * 6 + 2 * 5 + 2 * 4 + 4 * 3 + 7 * 2$$

O resultado do nosso exemplo é:

295

O próximo passo da verificação também é simples, basta multiplicarmos esse resultado por 10 e dividirmos por 11.

$$295 * 10 / 11$$

O resultado que nos interessa na verdade é o RESTO da divisão. Se ele for igual ao primeiro dígito verificador (primeiro dígito depois do '-'), a primeira parte da validação está correta.

**Observação Importante:** Se o resto da divisão for igual a 10, nós o consideramos como 0.

Vamos conferir o primeiro dígito verificador do nosso exemplo:

O resultado da divisão acima é '268' e o RESTO é 2

Adaptado de:  
<https://dicasdeprogramacao.com.br/algoritmo-para-validar-cpf/>

# Validador de CPF

## Validação do segundo dígito

A validação do segundo dígito é semelhante à primeira, porém vamos considerar os 9 primeiros dígitos, mais o primeiro dígito verificador, e vamos multiplicar esses 10 números pela sequência decrescente de 11 a 2. Vejamos:

$$5 * 11 + 2 * 10 + 9 * 9 + 9 * 8 + 8 * 7 + 2 * 6 + 2 * 5 + 4 * 4 + 7 * 3 + 2 * 2$$

O resultado é:

$$347$$

Seguindo o mesmo processo da primeira verificação, multiplicamos por 10 e dividimos por 11.

$$347 * 10 / 11$$

Verificando o RESTO, como fizemos anteriormente, temos:

$$\text{O resultado da divisão é '315' e o RESTO é 5}$$

Verificamos, se o resto corresponde ao segundo dígito verificador.

Com essa verificação, constatamos que o CPF 529.982.247-25 é válido.

Adaptado de:  
<https://dicasdeprogramacao.com.br/algoritmo-para-validar-cpf/>

## CPFS INVÁLIDOS CONHECIDOS

Existe alguns casos de CPFs que passam nessa validação que expliquei, mas que ainda são inválidos.

É os caso dos CPFs com dígitos repetidos (111.111.111-11, 222.222.222-22, ...)

Esses CPF atendem à validação, mas ainda são considerados inválidos.

No nosso algoritmo, vamos verificar se todos os dígitos do CPF são iguais e, neste caso, considerar que ele é inválido.

Adaptado de:  
<https://dicasdeprogramacao.com.br/algoritmo-para-validar-cpf/>

# Perguntas?