

Java EFETIVO → ler Quando Finalizar o curso  
Codigo Limpo → //

Um Princípio muito importante do código Limpo, é nomear de uma forma melhor as coisas, ou seja, escolher bons nomes.

Ex: Variáveis, classes, métodos, etc...

"Programas São Feitos Para serem lidos Por Humanos e apenas incidentalmente Para serem executados por computadores."

- Donald Knuth

"Existem apenas duas coisas difíceis na ciência da computação: invalidar cache e dar nome às coisas."

- Phil Karlton

⇒ Se o nome de alguma coisa precisa de um comentário para explicar, então você já sabe que o nome não é bom o suficiente p/ revelar a intenção.

⇒ Usar nomes claros

⇒ Numero Solto = Atribuir a uma constante

⇒ Nome de classe não deve ser um verbo, deve ser substantivo e, uma coisa

⇒ Usar nomes descritivos, Não precisa ter medo de usar nomes longos.

⇒ Sempre que ver os códigos Fugindo desses princípios, repatore.

## Organização da classe:

- 1:) Variáveis de classe
- 2:) Variáveis de instâncias
- 3:) métodos.

## Tamanho da classe:

- 1:) classes devem ser pequenas
- 2:) menor do que você pensou

O tamanho da classe deve ser medido de acordo com a quantidade de **responsabilidade**.

\* Se você está com dificuldades de dar nome conciso a sua classe, talvez ela tenha responsabilidades demais. O nome da classe deve descrever a sua responsabilidade.

\* Principio: Single responsibility Principal

↳ Principio de responsabilidade unica

↳ SRP

ideal → Sem comentário.

tentar deixar o código o mais legível  
possível para que não exista comentários.

## Princípios de código limpo Para Métodos:

O quão Pequeno deve ser um método?

→ 1 linha

→ Métodos devem fazer uma única coisa.

Como identificar se um método está fazendo mais coisas?

→ Verificando se dá para extrair um outro método, A partir desse método original com um nome que faça sentido.

Qual o número ideal de argumentos?

→ 0

→ o ideal é que os seus métodos não tenham nenhum argumento.

→ Você pode criar uma classe que representa a coisa que você tá desenvolvendo, você pode ter variáveis de instâncias, e você pode ter métodos que trabalham em cima dessas instâncias.

→ Mais que 3 argumentos, certamente deve ter alguma forma de melhorar isso.

→ Usar mais objetos como parâmetro



Princípio: Você nunca deve passar null para um argumento de um método para indicar que não tem nenhum valor para esse argumento.

⇒ Para validar se um objeto não é null, basta chamar

`Objects.requireNonNull(object)`

↳ ele lança uma exceção