

Programação Orientada a Objeto

OO3 - Atributos e Métodos

Dr. Márcia L. Aguená Castro

marciaaguená@gmail.com

- Um **atributo** representa um **valor** que um objeto deve possuir, ou um **estado** que o objeto apresenta:
 - Ex: uma pessoa possui nome, idade, é casada, etc
- Um **método** é uma função que exprime um **comportamento** de um objeto:
 - Ex: uma pessoa fala, anda, etc

Atributos

- Tanto os atributos quanto os métodos devem fazer sentido **em relação ao domínio do problema** em que o software está inserido.
 - O atributo cor dos olhos de uma pessoa:
 - Faz sentido em um sistema contabilidade?
 - E em um sistema de controle hereditário?
 - O que muda entre as situações?

- Os métodos mais comuns implementados em todas as classes são os métodos `get's` e `set's`;
- Os métodos iniciados com `get's` retornam um valor para **fora** da classe:
 - Exemplo: `getNome()`, `getValor()`, `get<insira seu atributo aqui>`.
- Sempre têm com o **tipo de retorno** o mesmo do **atributo** tratado.

OO3 - Abstração, Classes e Objetos, Atributos e Métodos

Atributos e Métodos

- Os métodos iniciados com set's mudam o valor de um atributo;
- Colocam um valor em um atributo **dentro** da classe:
 - Exemplo: `setNome()`, `setValor()`,
`set<insira seu atributo aqui>`.
- Sempre têm com o tipo de retorno `void`, ou seja, **não retornam valor**.
- Veremos o porquê da popularidade desses métodos logo mais.

OO3 - Abstração, Classes e Objetos, Atributos e Métodos

Atributos e Métodos

```
1 public class Pessoa{  
2     public String nome;  
3     public String getNome() {  
4         return this.nome;  
5     }  
6 }
```

Declaração de um atributo
nome e um método getNome()
na classe Pessoa

Utilização do atributo e do
método na classe
TestaPessoa

```
1 public class TestaPessoa{  
2     public static void main (String[] args) {  
3         Pessoa p1 = new Pessoa();  
4         p1.nome = "Reinaldo";  
5         System.out.println("O nome da pessoa é " + p1.getNome());  
6     }  
7 }
```

OO3 - Abstração, Classes e Objetos, Atributos e Métodos

Atributos e Métodos

- Os métodos iniciados com set's mudam o valor de um atributo;
- Colocam um valor em um atributo **dentro** da classe:
 - Exemplo: `setNome()`, `setValor()`,
`set<insira seu atributo aqui>`.
- Sempre têm com o tipo de retorno `void`, ou seja, **não retornam valor**.
- Veremos o porquê da popularidade desses métodos logo mais.

OO3 - Abstração, Classes e Objetos, Atributos e Métodos

Dicas de Qualidade de Código

- Uma classe deve implementar **somente um** TAD;
- Quando isso ocorre, o programador atinge maior **coesão** em seu código;
- Você percebe a **coesão** do código por pistas que os métodos dessa classe deixam (desconsiderando get's e set's):
 - Dado um método de uma classe, ele deve trabalhar com **vários atributos dessa classe**, e não somente com um pequeno subconjunto desses.

OO3 - Abstração, Classes e Objetos, Atributos e Métodos

Atributos e Métodos - Exercício

```
1 public class Paciente {
2     String nome;
3     int idade;
4     String rua;
5     int numero;
6     String bairro;
7     String cidade;
8     String estado;
9     String cep;
10
11     //get's e set's omitidos
12
13     public void alteraEndereco(String $nome, int $idade, String $rua ,int $numero,
14                               String $bairro, String $cidade, String $estado, String $cep){
15         this.rua      = $rua;
16         this.numero = $numero;
17         this.bairro = $bairro;
18         this.cidade = $cidade;
19         this.estado = $estado;
20         this.cep     = $cep;
21     }
22 }
```

Altere a classe
Paciente de forma que
ela fique mais coesa.

A palavra-chave **this**
refere-se ao objeto que
chamou o método.

OO3 - Abstração, Classes e Objetos, Atributos e Métodos

Atributos e Métodos - Exercícios

Atenção: Não copie ou cole nenhum exercício. A repetição é intencional para criar fluência na linguagem.

Observação: Guarde o diretório sistema, pois incrementaremos as classes com cada conceito de OO aprendido durante esse treinamento.

1. Abra a classe Banco que você criou anteriormente e adicione os seguintes atributos:

- "id" do tipo long,
- "numero" do tipo String,
- "cnpj" do tipo String,
- "nome" do tipo String;

OO3 - Abstração, Classes e Objetos, Atributos e Métodos

Atributos e Métodos - Exercícios

2. Ainda na classe `Banco`, crie métodos `get's` para cada atributo declarado.
3. Abra a classe `Agencia` que você criou anteriormente e adicione os seguintes atributos:
 - `"id"` do tipo `long`
 - `"numero"` do tipo `String`
 - `"nome"` do tipo `String`
4. Ainda na classe `Agencia`, crie métodos `get's` para cada atributo declarado.

OO3 - Abstração, Classes e Objetos, Atributos e Métodos

Atributos e Métodos - Exercícios

5. Abra a classe Conta que você criou anteriormente e adicione os seguintes atributos:
 - "id" do tipo long
 - "numero" do tipo String
 - "saldo" do tipo double
6. Ainda na classe Conta, crie métodos get's para cada atributo declarado.

OO3 - Abstração, Classes e Objetos, Atributos e Métodos

Atributos e Métodos - Exercícios

7. Dentro do diretório `sistema`, crie a seguinte classe pública:

- `ContaCorrente`, com os atributos:
 - Todos os atributos de `Conta` ("`id`", "`numero`" e "`saldo`");
 - "`limite`" do tipo `double`;

8. Ainda na classe `ContaCorrente`, crie métodos `get`'s para cada atributo declarado.

9. Complemente o método `main` da classe `AplicacaoFinanceira` com as seguintes operações:
- Crie o objeto `contaCorrente1` da classe `ContaCorrente`;
 - Inicialize todos os atributos dos objetos `banco1`, `agencia1`, `conta1` e `contaCorrente1` com valores compatíveis com seus tipos;

10. Complemente o método `main` da classe

- Imprima o valor dos atributos `"id"` e `"numero"`, via `System.out.println()` e os métodos `get's`, para os 3 primeiros objetos do passo anterior e `"id"` e `"limite"` para o objeto `contaCorrente1` ;

11. Compile todas as classes para verificar se existe algum erro.