

Programar em Linguagem Orientada a Objetos Básica

Prof. Arthur M. Araújo

Programação Orientada a Objetos (POO)

- Conceito POO
- Classes e Objetos
- Atributos
- Métodos
- Modificadores de acesso
- Exercício



Resolvendo um problema

Vamos criar um programa para calcular a área de um terreno. Primeiro, vamos ler as medidas de largura e comprimento do terreno. Em seguida, mostraremos o valor da área em metros quadrados.

A fórmula para calcular a área do terreno é obtida multiplicando-se a largura pelo comprimento.

$$\text{area} = \text{largura} * \text{comprimento}$$

Exemplo de Execução:

```
Digite a largura do terreno: 10  
Digite o comprimento do terreno: 15  
A área do terreno é: 150 metros quadrados.
```

Conceito de POO

O conceito central da **Programação Orientada a Objetos (POO)** é projetar e estruturar software em torno de objetos, que são **instâncias** de classes. A POO é um paradigma de programação que organiza o código em torno de **objetos**, que podem conter **dados** (atributos) e **funcionalidades** (métodos) relacionadas a um conceito ou entidade específica.

Os principais princípios da POO incluem:

- Encapsulamento
- Herança
- Polimorfismo

POO: Classes

Classes:

- Uma classe é um **modelo** ou um **plano** para criar **objetos**. Ela define a estrutura e o comportamento dos objetos que serão criados a partir dela.
- Uma classe contém **atributos** (variáveis de instância) para representar os dados e **métodos** (funções) para definir o comportamento dos objetos.
- As classes em Java são definidas usando a palavra-chave **class**.

Exemplos:

- `public class Pessoa{};`
- `public class Tarefa{};`
- `public class Produto{};`
- `public class Calculadora{};`

POO: Objetos

Objetos:

- Um **objeto** é uma **instância** de uma classe. Ele é criado com base no **modelo** definido pela classe e possui seus próprios valores de **atributos**.
- Os objetos são usados para representar **entidades** do mundo real no código, permitindo a manipulação de **dados** e a execução de ações específicas.
- Para criar um objeto a partir de uma classe em Java, você usa a palavra-chave **new**.

Exemplos:

- **new** Pessoa();
- **new** Produto();
- **new** Tarefa();

POO: Classes e Objetos

A imagem mostra uma **planta** de uma casa, representando a **classe**, e uma casa já **construída**, representando o **objeto**. A classe é o modelo ou plano, enquanto o objeto é a instância concreta desse modelo. A partir dessa planta, é possível criar várias casas, cada uma sendo um objeto distinto.



= Objeto

A casa **construída** é a instância do modelo da casa abaixo.



= Classe

A **planta** da casa é um modelo de uma casa.

POO: Atributos

Atributos:

- Atributos, também conhecidos como **variáveis** de instância, são as variáveis associadas a uma classe.
- Eles representam os **dados** ou **características** de objetos criados a partir dessa classe.
- Cada objeto tem sua própria cópia dos **atributos** da classe.
- Os atributos **definem** o estado dos objetos.
- Eles podem ter diferentes tipos de dados, como **Integer**, **Strings**, **Boolean**, entre outros.

Exemplos:

- Pessoa: **nome**, **idade**, **altura**.
- Produto: **nome**, **preco**, **marca**.
- Tarefa: **descricao**, **dataInicio**, **prioridade**.

Revisão

Agora, utilizando a classe, temos uma variável composta do tipo "Terreno" para representar as dimensões de um terreno em largura e comprimento. Para isso, criamos duas instâncias da classe.

```
Terreno x, y;  
x = new Terreno();  
y = new Terreno();
```

A seguir, vamos aprimorar nossa classe, adicionando um método para calcular a área em metros quadrados.

POO: Métodos

Métodos:

- Métodos são **funções** associadas a uma **classe** e definem o **comportamento** dos objetos criados a partir dessa classe.
- Eles representam as **ações** que os objetos podem executar.
- Métodos podem ter **parâmetros**, que são **informações** que podem ser passadas para eles, e um tipo de **retorno**, que especifica o tipo de dado que o método retorna.
- Métodos podem ser chamados em objetos da classe para realizar operações específicas.

Exemplos:

- Carro:
 - *ligarMotor();*
 - *calcularConsumo();*

POO: Modificadores de Acesso

Os **modificadores de acesso** são palavras-chave que determinam o nível de visibilidade e acesso de classes, atributos e métodos em relação a outras classes. Eles são usados para controlar a encapsulação e a segurança no programa. Existem quatro principais modificadores de acesso em Java:

- **public (Público):** Os elementos (classes, métodos e atributos) marcados como public são acessíveis a partir de qualquer classe em qualquer pacote. Eles têm o maior nível de visibilidade.

POO: Modificadores de Acesso

- **private (Privado):** Os elementos marcados como private são acessíveis apenas dentro da classe em que são definidos. Eles não podem ser acessados de fora da classe, mesmo em subclasses.
- **protected (Protegido):** Os elementos marcados como protected são acessíveis dentro da mesma classe, em subclasses e em classes do mesmo pacote. Eles não são acessíveis fora do pacote a menos que haja herança.
- **Nenhum Modificador (Padrão):** Quando nenhum modificador de acesso é especificado, o elemento tem o modificador de acesso padrão, que é chamado de "package-private" ou "nível de pacote". Isso significa que o elemento é acessível apenas dentro da mesma classe e em classes do mesmo pacote.

Benefícios

Quais os benefícios de se calcular a área do terreno por meio de um **Método** dentro da **Classe Terreno**?

- **Reaproveitamento de código:** uma vez que você tenha definido uma classe com seus atributos e métodos, você pode criar várias instâncias dessa classe em seu programa, evitando a necessidade de redefinir a mesma estrutura repetidamente.
- **Delegação de responsabilidades:** o responsável por saber calcular a área do terreno é o próprio Terreno, ou seja, a lógica não deve estar em outro lugar.

Benefícios

- **Organização Estruturada:** Usar uma classe permite agrupar atributos relacionados (como as três notas que você mencionou) em uma única entidade, tornando o código mais organizado e fácil de entender.
- **Facilidade de Manutenção:** Ao usar objetos e classes, as alterações em um componente podem ser feitas em um único local, afetando todas as instâncias desse componente, o que simplifica a manutenção e reduz a possibilidade de erros.

Exercício - Calcular o IMC

Vamos criar um programa para calcular IMC de uma pessoa. Primeiro, vamos obter o nome, idade, altura e peso. Em seguida, mostraremos as informações da pessoa e o resultado do IMC.

A fórmula para calcular o IMC pela fórmula:

$$\text{IMC} = \text{peso} / (\text{altura} * \text{altura})$$

Dicas:

1. Implementar classe Pessoa com os atributos da mesma.
2. Implementar os métodos de imprimir detalhes da pessoa e calcular IMC.

Obrigado!

Contatos



Prof. Arthur M. Araújo



arthur.araujo@maisunifacisa.com.br