

Lista de exercícios I – Programação Orientada a Objetos

Marcela de Oliveira Araújo

1. Por que programação orientada a objetos tem esse nome? O que são objetos nesse contexto?

O nome **Programação Orientada a Objetos (POO)** vem da ideia central da abordagem: em vez de escrever programas apenas como uma sequência de instruções (como na programação procedural), você organiza o código em **objetos** que representam entidades do mundo real ou conceitos abstratos. Cada objeto "**possui**" **dados e comportamentos**.

Um **objeto** é uma **instância de uma classe**. Pense em uma classe como uma **planta ou modelo**, e o objeto como uma **casa construída a partir dessa planta**.

2. O que é uma instância?

Na Programação Orientada a Objetos, **instância** é o nome que damos quando **criamos um objeto a partir de uma classe**.

3. O que é uma classe?

Uma **classe** é um **molde ou modelo** que descreve como os **objetos** serão. Ela **define**: **Atributos** → as características (dados) que os objetos terão e **Métodos** → os comportamentos (ações) que os objetos poderão realizar

4. Para que serve um método construtor? Como ele é chamado (mostre um exemplo de código)?

Um **método construtor** serve para **inicializar** um objeto quando ele é criado.

Ele define **como o objeto começa**, atribuindo valores iniciais aos seus atributos ou executando alguma lógica necessária na criação.

5. Os métodos construtores podem ser vazios?

Sim! Isso significa que o construtor **não faz nada** além de permitir a criação do objeto.

6. Em Java, um programa tem método destrutor? Por quê?

~~Em~~ **Em Java, não existe destrutor**. Isso acontece porque:

O Java tem o **Garbage Collector (GC)**, um coletor de lixo que cuida automaticamente de liberar a memória de objetos que não estão mais sendo usados.

Ou seja, o programador **não precisa (e nem consegue)** controlar diretamente a **destruição de objetos**.

7. Para que serve o Garbage Collector do Java? Cite uma vantagem do GC. Cite uma desvantagem.

O **Garbage Collector** é um mecanismo automático da JVM (Java Virtual Machine) que serve para **gerenciar a memória**.

Ele identifica e **remove da memória os objetos que não estão mais sendo utilizados** pelo programa (ou seja, que não têm mais referências ativas). Assim, evita **vazamentos de memória** e facilita a vida do programador, que não precisa liberar a memória manualmente (como em C/C++).

8. O que são bytecodes e máquina virtual Java? Apenas a linguagem Java pode gerar bytecodes?

O **bytecode** é um **código intermediário** gerado pelo compilador Java (javac).

Em vez de compilar diretamente para **código de máquina** (específico de cada processador e sistema operacional), o Java compila para bytecode. Esse bytecode é **independente de plataforma** e pode rodar em qualquer sistema que tenha a **JVM (Java Virtual Machine)**.

9. Cite os fundamentos básicos da Programação Orientada a Objetos e defina dois deles.

Os 4 fundamentos básicos da Programação Orientada a Objetos:

1. Abstração
2. Encapsulamento
3. Herança
4. Polimorfismo

10. O que é o tipo var no Java?

O **var** foi introduzido no **Java 10** e serve para **inferência de tipo local**. Isso significa que o compilador Java **descobre automaticamente o tipo da variável** a partir do valor atribuído a ela.