



Programação Java

➤ Introdução à linguagem Java

// Prática Integradora - Programação Orientada a Objetos

Objetivo

O objetivo deste guia prático é que possamos consolidar e aprofundar os conceitos de programação orientada ou objetos aplicados a Java. Para isso vamos propor uma série de exercícios simples e incrementais (já que vamos trabalhar e adicionar lógica às aulas que temos que construir), que nos permitirão revisar os tópicos que estamos estudando.

Are you ready? 😎👍



Exercício 1

Crie uma classe Pessoa, que terá os seguintes atributos: nome, idade, id (neste caso, vamos representá-la como uma string de caracteres), peso e altura. Que tipo de dados você atribuiria às variáveis de instância? Qual seria a estrutura básica de sua classe?



Exercício 2

Vamos criar diferentes construtores na classe Pessoa, um sem parâmetros, o segundo deve receber nome, idade e ID como parâmetro; finalmente criamos um terceiro que recebe como parâmetro todos os atributos da classe.



Exercício 3

Crie uma nova classe chamada Main, onde você declara um método principal como ensinamos anteriormente. Isso nos permitirá executar nosso aplicativo.



Exercício 4

Na classe Main que acabamos de criar, dentro do método main () pedimos que você crie um objeto do tipo Pessoa para cada construtor que definimos na classe, lembre-se de colocar um nome significativo para as variáveis para onde você está indo atribuir cada objeto. Como você faria? Em seguida, vamos criar outro objeto do tipo pessoa e vamos construí-lo passando apenas um valor para o nome e outro para a idade no método construtor. Isto é possível? O que acontece se tentarmos fazer isso?



Exercício 5

Na classe Pessoa iremos implementar os seguintes métodos: `calcularIMC()`, a fórmula para calcular é: $\text{peso} / (\text{altura}^2)$ onde peso expresso em kg e altura em metros. Se este cálculo retornar um valor menor que 20, a função deve retornar -1. Se retornar um número entre 20 e 25 **inclusive para os dois valores**, o método deve retornar 0. Por fim, se retornar um número maior que 25 deve retornar 1. Uma vez que o método anterior foi criado, vamos adicionar o método `eMaiorDeldade()` que deve retornar um valor booleano, levando em consideração que a maioridade será considerada neste caso, a partir dos 18 anos. Por fim, adicione um método `toString()` que retornará todas as informações da pessoa.

Lembrar! Você pode se ajudar com os métodos da classe `java.lang.Math` para calcular a potência.



Exercício 6

A partir da classe Main vamos calcular o IMC da última pessoa que criamos (aquela que criamos corretamente usando o construtor que recebe todos os atributos como parâmetro). Também saberemos se você é maior de idade ou não; Lembre-se que em ambos os casos, dependendo dos resultados retornados pelos métodos, deve-se imprimir uma mensagem correspondente para o usuário. Por fim, queremos mostrar todos os dados dessa pessoa, imprimindo essas informações por console. O formato em que você vai mostrar os dados e as mensagens fica a seu critério, mas deve ser legível e descritivo para quem vê a saída do programa.

Referencias:

| Índice de massa corporal (IMC) | Nível de peso |
|--------------------------------|---------------|
| Abaixo de 20 | Baixo peso |
| Entre 20 e 25 | Peso saudável |
| Acima de 25 | Sobrepeso |