

Exercício 1

a) O que é uma classe?

Classe é a estrutura fundamental de uma linguagem de programação orientada a objetos. Ela é a abstração de uma entidade ou objeto do mundo real, cuja finalidade é representar suas características e comportamentos mediante atributos e métodos.

b) O que é um objeto?

Um objeto é a instância de uma classe, isto é, um elemento particular que contém todas as características definidas por uma classe, de modo que os atributos representam as características específicas ou o estado atual do objeto e os métodos realizam operações sobre ele.

c) O que é um atributo?

Atributo é um dos componentes de uma classe, cuja finalidade é representar uma característica ou um estado de um objeto instanciado por essa classe.

d) O que é um método?

Método é um dos componentes de uma classe, que consiste em uma função que permite realizar ações ou alterar o estado de um objeto instanciado por essa classe.

e) O que é a assinatura de um método?

A assinatura de um método, conforme algumas literaturas, é composta pelo nome do método, tipo de retorno (`int`, `double`...), visibilidade (`public`, `private`...), quantidade de parâmetros, tipos de parâmetros (`int`, `double`...) e a ordem deles. Em Java, quando se trata de sobrecarga de métodos, leva-se em conta a assinatura como sendo a quantidade, os tipos e ordem dos parâmetros.

f) O que é um construtor e quando ele é invocado?

Um construtor é um método de uma classe que é executado no momento em que um objeto é instanciado. É normalmente utilizado para definir atributos padrões, alocar variáveis etc.

g) Como eu seleciono o construtor que será invocado?

Na criação de um objeto, seleciona-se o tipo de construtor mediante a assinatura utilizada, isto é, conforme a quantidade e a ordem dos tipos de parâmetros.

h) Para que serve o método main?

O método **main** representa a função principal de uma classe. Ao executar o programa dessa classe, a primeira função que será chamada é a **main**. Sendo assim, sua finalidade é dar início a uma série de comandos, operações e chamadas de outras funções, de modo a cumprir a tarefa atribuída ao programa.

i) O que significa a sobrecarga de métodos?

Sobrecarga de métodos é o caso em que há dois ou mais métodos definidos com o mesmo nome. Nessa situação, é necessário que tais métodos tenham assinaturas distintas no que diz respeito à quantidade e/ou ordem dos tipos de parâmetros.

j) O que são atributos e métodos estáticos (static) e para que eles servem?

Atributo estático é um atributo comum a todas as classes, o que implica que qualquer alteração nele produz efeito em todos os objetos dessa classe. Sua finalidade é armazenar informações que se aplicam a todos os objetos simultaneamente, simulando a funcionalidade de uma variável global.

Métodos estáticos são métodos que não precisam de objetos para sua utilização, ademais, só podem acessar atributos ou métodos de classe, mas nunca atributos ou métodos de instância. Sua finalidade é descrever as funções que não necessitam de uma instanciação para exercerem sua operação.

Tanto os atributos quanto os métodos estáticos são precedidos pelo modificador **static**, e, além disso, são resolvidos em tempo de compilação, por não serem dinâmicos.

k) Qual a diferença entre um atributo de instância e um atributo de classe?

Atributo de instância é um atributo que pertence a um objeto criado, e sua alteração não produz efeito em nenhum outro objeto a não ser ele próprio.

Atributo de classe, por sua vez, é um atributo comum a todas as classes, análogo a uma variável global. Sua alteração produz efeito em todos os objetos dessa classe e ele pode ser acessado através do nome da classe. Um atributo estático, como também é chamado o atributo de classe, é precedido pelo modificador **static**.

l) Para que servem os modificadores de acesso?

Os modificadores de acesso têm a finalidade de definir o nível de visibilidade de uma classe, atributo ou método, ou seja, o quão restrito é o seu acesso. Existem quatro palavras-chave reservadas pelo Java que servem como modificadores de acesso, são elas:

public: pode ser acessado por qualquer classe.

protected: apenas a própria classe e as subclasses (filhas) podem acessá-lo.

private: somente a própria classe pode acessá-lo.

Por padrão, se nenhum desses modificadores for utilizado, então o acesso é limitado às classes do mesmo pacote.

Exercícios de implementação:

Exercício 2: implementado completamente (projeto "Exercise2").

Exercício 3: implementado completamente (projeto "Exercise3").

Exercício 4: implementado completamente (projeto "Exercise4").

Exercício 5: implementado completamente (projeto "Exercise5").