

Programação Orientada a Objetos I

Você acessou como **João Manoel Leite Ribeiro Nogueira** (Sair)

[Página inicial](#) > [Meus cursos](#) > [POOI](#) > [Introdução a Classes e Objetos](#) > [Tarefa para Casa: Revisão dos Conceitos OO](#)

Conforme o exemplo a seguir, determine se cada uma das afirmações é verdadeiro ou falso.

Exemplo de Afirmação:

- *(a) As variáveis declaradas no corpo de um método particular são conhecidas como variáveis de instância e podem ser utilizadas em todos os métodos da classe.*
- *(b) As variáveis ou métodos declarados com o modificador de acesso private só são acessíveis a métodos da classe em que são declarados.*
- *(c) Por convenção, os nomes de método iniciam com a primeira letra maiúscula e todas as palavras subsequentes do nome iniciam com a primeira letra maiúscula.*

Exemplo de Resposta:

Navegação



Página inicial

▪ Minha página inicial

Páginas do site

Meu perfil

Meus cursos

POOI

Participantes

Geral

Materiais da Disciplina

Introdução

Introdução a Classes e Objetos

Classes x Objetos

Tarefa para Casa: Revisão dos Conceitos OO

Exemplo de Implementação: Dia 19/09

Exercício em Sala

Envio do Exercício em Sala

Exemplo de Implementação: Dia 26/09

Agregação e Composição

Avaliação: Agregação e Composição

Configurações



Administração de tarefas

- **(a) Falso.** *Essas variáveis são chamadas de variáveis locais e podem ser utilizadas somente no método em que são declaradas.*
- **(b) Verdadeiro.**
- **(c) Falso.** *Por convenção, os nomes de métodos iniciam com a primeira letra minúscula e todas as palavras subsequentes no nome iniciam com a primeira letra maiúscula.*

Utilizando o padrão de resposta acima, analise as questões abaixo, determinando se cada uma das afirmações é verdadeiro ou falso. Se falso, explique por quê.

- **(a)** O paradigma de programação estruturada visualiza e representa o mundo real como um conjunto de objetos que interagem entre si para que determinadas operações sejam realizadas.
- **(b)** Uma casa está para uma planta arquitetônica assim como um modelo está para uma classe.

- [Editar o documento enviado](#)

Administração do curso

Minhas configurações de perfil

- **(c)** O número de argumentos na chamada de um método deve corresponder ao número de parâmetros na lista de parâmetros da declaração de um método.
- **(d)** Na POO, uma classe é uma abstração de entidades existentes no domínio do sistema de software.
- **(e)** Um programa orientado a objetos é composto por um conjunto de objetos que interagem entre si.
- **(f)** Toda declaração de classe que inicia com a palavra-chave `private` e `protected` deve ser armazenada em um arquivo que tem exatamente o mesmo nome que a classe e terminar com a extensão de nome do arquivo `.java`.
- **(g)** A unidade fundamental de programação em orientação a objetos é procedimento.
- **(h)** Uma mensagem é um elemento usado para instanciação de de objetos.

- **(i)** A classe é uma coleção de dados (atributos) e operações (métodos) que manipulam tais dados.
- **(j)** Em A/POO, a abstração permite que uma mesma operação possa ser definida para diferentes tipos de classes, e cada uma delas a implementa como quiser.
- **(k)** Métodos operam no estado interno de um objeto e servem como mecanismo de comunicação entre objetos.
- **(l)** O encapsulamento permite ignorar os detalhes de implementação permitindo ao desenvolvedor idealizar seu trabalho em um nível mais alto de abstração.
- **(m)** A herança ou generalização permite que elementos mais específicos incorporem a estrutura e o comportamento de elementos mais genéricos.
- **(n)** Em POO, o polimorfismo permite a construção de um

modelo para representação de uma realidade, com foco nos aspectos essenciais.

- **(o)** A interface da classe é conjunto de instâncias de uma classe, indicando como os objetos podem interagir através da troca de mensagens.
- **(p)** Um objeto de uma subclasse (classe filha) é um membro de objeto da superclasse (classe pai).
- **(q)** Os modificadores de acesso definem o escopo/visibilidade de um método/atributo.
- **(r)** O modificador de acesso "public" é simbolizado por "+" e indica que o atributo/método pode ser acessado por objetos de outras classes.
- **(s)** O modificador de acesso "private" é simbolizado por "#" e permite o acesso a objetos de classes filhas mas protege do acesso de objetos que não fazem parte da hierarquia de classes.
- **(t)** O modificador de acesso "protected" é simbolizado por "-" e

protege o atributo do acesso externo, permitindo ao mesmo ser acessado somente por métodos daquela classe e pelo objeto instanciado.

- **(u)** Aplicando a ideia do encapsulamento, os atributos deveriam ser todos privados.
- **(v)** Para realizar uma tarefa em um programa OO é necessário um método. Dentro do método é possível inserir os mecanismos para que eles façam suas tarefas.
- **(w)** Acessar ou modificar as propriedades de um objeto manipulando diretamente os seus atributos é uma abordagem recomendável, pois é um dos princípios da ideia do encapsulamento.
- **(x)** O corpo de todos os métodos é delimitado pelas chaves esquerda e direita ({ e }).
- **(y)** Na UML, cada classe é modelada em diagrama de classe como um retângulo com três compartimentos. Aquele da parte

superior contém as operações da classe. O do meio contém os atributos da classe. o inferior contém o nome da classe.

- **(z)** As classes costumam fornecer métodos public para permitir aos clientes da classe configurar (set) ou obter (get) variáveis de instância private. Os nomes desses métodos não precisam iniciar com set ou get, mas esta convenção de nomes é recomendada.

Disponível a partir de:	domingo, 18 setembro 2016, 18:15
Data de entrega:	domingo, 18 setembro 2016, 23:55

Você acessou como [João Manoel Leite Ribeiro Nogueira](#) ([Sair](#))

POOI