



INSTITUTO FEDERAL  
GOIÁS

INSTITUTO FEDERAL DE GOIÁS  
CÂMPUS GOIÂNIA  
BACHARELADO EM SISTEMAS DE INFORMAÇÃO  
PROGRAMAÇÃO ORIENTADA A OBJETO I

Nome do Aluno: João Manoel Leite Ribeiro Nogueira Data: 18/09/2016

Prof. Renan Rodrigues de Oliveira

## REVISÃO DOS CONCEITOS OO

- **(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.
- **(a) Falso.** *Esse paradigma é corretamente denominado de “programação orientada a objetos” e não de “programação estruturada”.*
- **(b)** Uma casa está para uma planta arquitetônica assim como um modelo está para uma classe.
- **(b) Falso.** *Uma casa está para uma planta arquitetônica assim como uma **classe** está para um **modelo**.*
- **(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.
- **(c) Verdadeiro.**
- **(d)** Na POO, uma classe é uma abstração de entidades existentes no domínio do sistema de software.
- **(d) Verdadeiro.**
- **(e)** Um programa orientado a objetos é composto por um conjunto de objetos que interagem entre si.
- **(e) Verdadeiro.**
- **(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’`.
- **(f) Falso.** *A afirmação é verdadeira apenas para as declarações de classe que estiverem no topo da hierarquia de classes do arquivo `‘.java’`.*

- **(g)** A unidade fundamental de programação em orientação a objetos é procedimento.
- **(g) Falso.** A unidade fundamental de programação em orientação a objetos é a **classe**.
- **(h)** Uma mensagem é um elemento usado para instanciação de objetos.
- **(h) Falso.** Uma mensagem é um elemento usado para **prover a comunicação entre objetos**.
- **(i)** A classe é uma coleção de dados (atributos) e operações (métodos) que manipulam tais dados.
- **(i) Verdadeiro.**
- **(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.
- **(j) Falso.** Em A/POO é o **polimorfismo** e não a **abstração** que 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.
- **(k) Verdadeiro.**
- **(l)** O encapsulamento permite ignorar os detalhes de implementação permitindo ao desenvolvedor idealizar seu trabalho em um nível mais alto de abstração.
- **(l) Verdadeiro.**
- **(m)** A herança ou generalização permite que elementos mais específicos incorporem a estrutura e o comportamento de elementos mais genéricos.
- **(m) Verdadeiro.**
- **(n)** Em POO, o polimorfismo permite a construção de um modelo para representação de uma realidade, com foco nos aspectos essenciais.
- **(n) Falso.** Em POO, é a **abstração** e não o **polimorfismo** que 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.
- **(o) Falso.** A interface da classe é o conjunto de **operações** 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).
- **(p) Falso.** Um objeto de uma subclasse (classe filha) é um **tipo** de objeto da superclasse (classe pai).

- **(q)** Os modificadores de acesso definem o escopo/visibilidade de um método/atributo.
- **(q) Verdadeiro.**
- **(r)** O modificador de acesso "public" é simbolizado por "+" e indica que o atributo/método pode ser acessado por objetos de outras classes.
- **(r) Verdadeiro.**
- **(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.
- **(s) Falso. Esse modificador de acesso é "protected" e não "private".**
- **(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.
- **(t) Falso. Esse modificador de acesso é "private" e não "protected".**
- **(u)** Aplicando a ideia do encapsulamento, os atributos deveriam ser todos privados.
- **(u) Verdadeiro.**
- **(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.
- **(v) Verdadeiro.**
- **(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.
- **(w) Falso. Essa abordagem não é recomendável, pois fere os princípios da ideia do encapsulamento.**
- **(x)** O corpo de todos os métodos é delimitado pelas chaves esquerda e direita ({ e }).
- **(x) Verdadeiro.**
- **(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.
- **(y) Falso. O retângulo da parte superior contém o nome da classe, o do meio contém os atributos da classe e o inferior contém as operações 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.
- **(z) Verdadeiro.**