

Questionário – Conceitos de Classes em Java

Aluno 1: Daniel Willians Ignacio de Souza

Data: 12 /09 / 2018

Observação: As questões devem estar condizentes com o conteúdo abordados nos artigos.

Segundo o autor dos artigos **Understanding Classes in Java (Part 1, Part 2 e Part 3)**, Justin Albano, responda as questões abaixo:

- 1) Segundo o autor, qual a definição usada para “Classe”? Exemplifique o conceito de *state* e *behavior* apresentados pelo autor.

O autor define classe como simplesmente uma representação de alguma entidade ou coisa que é composta de **estado** e **comportamento**,

O estado de uma classe é o que a classe tem, enquanto o comportamento de uma classe é o que a classe pode fazer

- 2) Qual o principal propósito de um *behavior* de uma Classe?

O estado de uma classe é os dados associados a ela, enquanto os comportamentos de uma classe são as ações que a classe pode executar. Por exemplo, podemos classificar um veículo simples dizendo que ele tem um nome de fabricante, nome de modelo e ano de produção associado a ele. Da mesma forma, podemos dizer que nosso carro simples pode acelerar e desacelerar. Portanto, podemos definir estado e comportamento usando a seguinte regra.

- 3) Como é chamado algo que permanece verdadeiro antes e depois da conclusão da implementação de um *behavior*? Exemplifique.

Uma invariante é uma regra de estado da classe e deve permanecer verdadeira antes e depois da conclusão de um comportamento.

Exemplo: Se estivermos em um carro e começarmos com uma velocidade atual de 0 e desacelerarmos? Em geral, a velocidade atual nunca deve cair abaixo de 0

- 4) O que é garantido em uma Classe, por meio da combinação de estados ocultos e *interfaces* externamente visíveis?

A redução da aparência de nossa classe para outras classes é chamada de interface de nossa classe. As interfaces são um dos conceitos mais importantes em qualquer linguagem orientada a objetos e fornecem os meios pelos quais outras classes interagem com nossa classe (ou seja, outras classes interagem com nossa classe por meio de sua interface).

A combinação do estado oculto e da interface visível externamente garante que nossa classe mantenha seu encapsulamento ou o acesso restrito às partes privadas da classe somente por meio das partes públicas.

- 5) Para a definição de construtores de Classes, o autor destaca que a sintaxe é muito similar à definição de *behavior*, exceto devido a dois ajustes pontuais. Quais são eles?

Para um construtor, usaremos a mesma notação como outros comportamentos, exceto com dois ajustes: (1) Nenhum tipo de retorno é especificado e (2) o nome do comportamento corresponde ao nome da classe. O primeiro ajuste é introduzido porque o construtor não pode retornar nenhum valor: ele é simplesmente chamado durante a instanciação do objeto. O segundo ajuste é instituído porque diferencia o comportamento de outros comportamentos gerais. Também nos permite instanciar objetos usando a seguinte notação:

Casa myHouse é nova casa ()

- 6) Quando uma instância de Objeto Java é criada em memória, em qual região da memória o mesmo é alocado? Quais as principais vantagens dessa alocação em relação à JVM?

Todos os Objetos são colocados na divisão heap da memória. Com um tipo de Objeto instanciado, a JVM consegue utilizar a mesma parte alocada da memória para executar comportamentos de outros Objetos da mesma classe, evitando uma nova alocação de memória para cada um deles.

- 7) Como surgiu o conceito de Classes e Objetos? Descreva em detalhes exemplificando, quando necessário.

O conceito de classe é uma extensão natural de um antigo paradigma de programação: Programação Processual. Antes da introdução de linguagens como Java, C ++, Python e outras linguagens, as linguagens procedurais (como C Programming Language, de Dennis Ritchie) eram o padrão de fato para o software. Essas linguagens tinham dois conceitos principais:

- (1) Agregados de diferentes variáveis de dados denominadas estruturas e
- (2) funções.

Simplificando, uma estrutura é uma entidade composta de membros de dados primitivos arbitrários ou outras estruturas. Por exemplo, se quiséssemos definir um endereço de correspondência em C, poderíamos criar a seguinte estrutura de dados:

- 8) Qual a relação de uso das palavras chaves **this** ou **super** na implementação de um Construtor em Java?

Super é semelhante à palavra chave this, mas, em vez de representar o objeto atual, ele representa o objeto atual da superclasse.

- 9) Segundo o autor, o que são Interfaces em Java?

Interfaces estão no centro de qualquer linguagem orientada a objetos, incluindo Java. Eles representam as janelas pelas quais permitimos voluntariamente que o mundo exterior veja nossas aulas. Assim como em nossa notação anterior, uma interface é composta de declarações de método sem corpo ou campos associados. Por exemplo, se quisermos criar uma interface para um animal, poderíamos criar o seguinte:

```
public interface Animal {  
    public int getAge ();  
    public String makeNoise ();  
}
```

- 10) Segundo o autor, o que são Classes Abstratas em Java?

Antes de podermos definir o que é uma classe, primeiro precisamos entender por que precisamos deles para começar. O objetivo de qualquer sistema de software é modelar as coisas no mundo real, físicas e abstratas, e realizar algum trabalho útil usando esse modelo. Por exemplo, podemos criar um sistema de software baseado em nuvem modelando os servidores (entidades físicas) e os canais e conexões de rede (conceitos abstratos) e os dados de fluxo por meio desse modelo. Sem um mecanismo pelo qual capturar esses conceitos em representações discretas e agilizar as interações entre essas representações, não podemos resolver um problema ou completar um objetivo desejado de maneira sistemática (algorítmica).

Seguindo essa premissa básica, nós, como desenvolvedores, precisamos usar técnicas e linguagens que facilitem esse processo de design e modelagem. O paradigma orientado a objetos, em particular, baseia-se na captura de entidades físicas e abstratas em classes . Usando este mecanismo de captura simples, quase

qualquer idéia ou conceito concebível pode ser capturado e, mais importante, feito para fazer um trabalho útil.