**C206 – B (LISTA TEÓRICA – L2)**

**MATHEUS HENRIQUE MARTINS – 1445**

**09/09/2021**

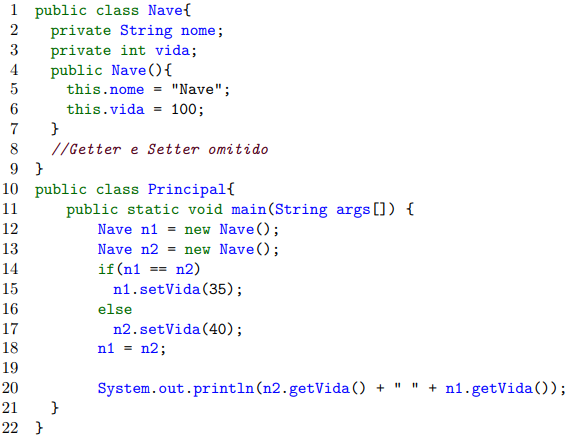
Considere o código apresentado na Figura 1 para responder as Questões 1 – 6

Figura 1

**Questão 1** O que será impresso no terminal ao executar esse código?

**R-** 40 40

**Questão 2** Explique o que as variáveis n1 e n2 significam. O que elas possuem salvo na memória?

**R-** As variáveis n1 e n2 significam novas instâncias de Nave. Elas possuem uma referência para esse objeto Nave.

**Questão 3** Qual o resultado do teste condicional da linha 14? Justifique sua resposta.

**R-** O teste condicional tem resultado falso. As variáveis n1 e n2 tem valores diferentes. n1 guarda uma referência e n2 guarda uma outra referência.

**Questão 4** O que está ocorrendo na linha 18? Qual o resultado dessa atribuição? Justifique sua reposta!

**R-** n1 está recebendo n2, ou seja, n1 recebe a referência que n2 guarda. Como resultado, ambas as variáveis estão referenciando o mesmo objeto.

**Questão 5** Qual o valor dos membros da classe Nave após a inicialização?

**R-** Após a inicialização, foi definido e atribuído pelo construtor os valores dos membros da classe Nave. O membro nome tem como valor a string "Nave" e o membro vida tem como valor o inteiro 100.

**Questão 6** O que significa o operador “this”? O que this.nome significa? Nesse exemplo, dentro do construtor da Nave, faz alguma diferença usar ou não o this? Justifique!

**R-** O operador “this” significa que você quer diferenciar quando a variável de escopo (ou parâmetro) tem o mesmo nome que o membro (ou variável da classe). E que você quer alterar apenas o membro daquela instância e não da classe.

***this.nome*** significa um acesso ao membro nome da classe Nave.

No exemplo, não faz diferença usar ou não o “this”, pois o método não recebe parâmetros. Então, não há conflito entre a variável de escopo e um membro da classe.

Sendo assim, não há necessidade do “this” para diferenciá-los.

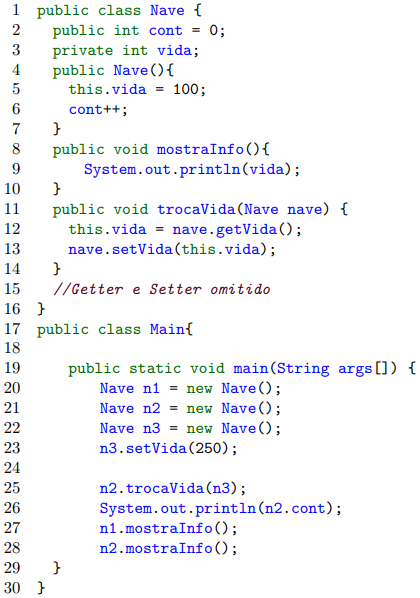
Considere o código apresentado na Figura 2 para responder as Questões 7 - 10. Considere também que não existe erro de compilação.

Figura 2:

**Questão 7** O que será impresso na tela ao executar o código na linha 26? Justifique esse valor impresso.

**R-** 1, pois a variável *cont* não foi definida como um modificador *static*. Sendo assim, ela será incrementada uma única vez. Logo, seu valor é 1.

CADA INSTANCIA TEM A SUA VARIAVEL CONT

**Questão 8** A variável membro “vida” (linha 3) está com o modificador private. O que isso significa? Seria possível fazer na classe Main uma chamada System.out.println(n1.vida)? Caso não seja possível, quais alternativas permitem ter acesso ao conteúdo da variável “vida”? Justifique sua resposta.

**R-** Isso significa que essa variável é privada. Os membros da classe definidos como *private* não podem ser acessados ou usados por nenhuma outra classe.

Então, não seria possível fazer essa chamada na classe Main, pois 'vida' tem acesso privado em 'Nave'. Como alternativa, podemos usar os métodos *getters* (acessores) e *setters* (modificadores) da nossa classe, que manipulam a variável de instância.

**Questão 9** A variável membro cont está sendo incrementada? Explique o funcionamento dessa variável. Qual o valor máximo que ela assume nesse código?

**R-** Sim, a variável membro *cont* está sendo incrementada, mas apenas uma vez. Cada instância tem o seu próprio *cont*, assim esse número sempre será incrementado uma única vez. Logo, seu valor máximo é 1.

Essa variável está funcionando como um contador, mas que pertence a instancia e não a classe. Por isso eles são chamados membros da instância.

**Questão 10** Seria possível no método Main executar o código Nave n4 = new Nave("Nave1")? Justifique.

**R-** Não, pois essa linha de código está criando/iniciando a variável n4 passando um parâmetro, e o construtor Nave na classe Nave não espera receber esse parâmetro para ser instanciado.

**Questão 11** O que significa comportamento da classe? E estado da classe? Dê exemplos.

**R-** Os membros indicam as possíveis informações armazenadas por um objeto de uma classe, representando o estado de cada objeto.

Os métodos são procedimentos que formam os comportamentos e ações oferecidos por objetos de uma classe, sendo responsáveis por alterar o estado ou fornecer informações sobre um objeto.

**Questão 12** O que é o construtor da classe? Quando ele é invocado? Ele pode ser parametrizado? Ele precisa ficar explícito? Justifique.

**R-** O construtor é o responsável por criar o objeto em memória, ou seja, instanciar a classe que foi definida. O construtor é unicamente invocado no momento da criação do objeto através do operador new.

Os construtores podem receber parâmetros (facilitando a inicialização de instâncias) e não precisam ficar explícito. Quando não declaramos o construtor explicitamente, o Java invoca o construtor implícito.