Livia Jacklinne Ramos Moreira, GEC - 1731

Lista de Exercícios 2 - Programação Orientada a Objetos

# Orientações:

* Entrega deve ser realizada via Tarefa do Microsoft Teams
* Entregar as respostas em arquivo formato PDF com identificação (Nome, Curso e Matrícula.)

# Objetivos da Lista:

* Diferença entre classe e objeto
* Diferença entre variáveis primitivas e variáveis de referência
* Construtores, modificadores de acesso e *static*

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

1. public class Nave{
2. private String nome;
3. private int vida;
4. public Nave(String nome, int vida){
5. this.nome = "Nave";
6. this.vida = 70;

7 }

8 *//Getter e Setter omitido*

9 }

10 public class Principal{

|  |  |  |
| --- | --- | --- |
| 11 |  | public static void main(String args[]) { |
| 12 |  | Nave n1 = new Nave("Nave1", 100); |
| 13 |  | Nave n2 = new Nave("Nave2", 200); |
| 14 |  | if(n1 != n2) |
| 15 |  | n1.setVida(50); |
| 16 |  | else |
| 17 |  | n2.setVida(70); |
| 18 |  | n2 = n1; |
| 19 |  |  |
| 20 |  | System.out.println(n2.getVida() + " " + n1.getVida()); |
| 21 | } |  |
| 22 } |  |  |

Figura 1:

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

**R:** Irá sair o valor da vida das naves: 50 50

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

**R:** n1 e n2 são dois objetos diferentes, eles tem salvo na memória o endereço de suas naves.

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

**R:** A resposta do teste nesse caso será TRUE, poisn1 e n2 são dois objetos, e cada um deles guarda um endereço.

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

**R:** O objeto, nave 2, está recebendo o endereço da nave 1. Portanto ao final da atribuição, tanto n1 quanto n2 apontarão para o mesmo endereço.

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

**R:** Nome Nave eVida 70

**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 “this” serve para dizer que é aquela variavel que será atribuida. O “this.nome”, está falando que esse nome que vai receber os dados. Sim, faz diferença. Primeiro o código não iria rodar; segundo, tem 3 atributos com o mesmo nome, então o compilador não saberia identificar o correto para fazer a atribuição.

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

1. public class Nave {
2. public static int cont = 0;
3. public int vida;
4. private int numNaves;
5. private String nome;
6. public Nave(){
7. this.vida = 100;
8. cont++;

9 }

1. public Nave(String nome){
2. this.nome = nome;
3. this.numNaves++;

13 }

1. public void mostraInfo(){
2. System.out.println(vida);

16 }

1. public void trocaVida(Nave nave) {
2. this.vida = nave.getVida();
3. nave.setVida(this.vida);

20 }

21 *//Getter e Setter omitido*

22 }

23 public class Main{

24

25 public static void main(String args[]) {

|  |  |  |  |
| --- | --- | --- | --- |
| 26 |  | Nave n1 = new Nave(); |  |
| 27 |  | Nave n2 = new Nave(); |  |
| 28 |  | Nave n3 = new Nave(); |  |
| 29 |  | n3.setVida(250); |  |
| 30 |  |  |  |
| 31 |  | n2.trocaVida(n3); |  |
| 32 |  | System.out.println(n2.cont); |  |
| 33 |  | n1.mostraInfo(); |  |
| 34 |  | n2.mostraInfo(); |  |
| 35 | } |  |  |
| 36 } |  |  | Figura 2: |

**Questão 7** O que será impresso na tela ao executar o código na linha 32? Justifique esse valor impresso.

**R:** 3, pois uma nova vida foi instanciados para tres novos objetos. Apesar de estarmos acessando “n2”, a variavel cont é STATIC, então ela tem seu valor alterado sempre.

**Questão 8** A variável membro “vida” (linha 3) está com o modificador public. 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:** Significa que ela pode ser alterada “fora” da classe, sem usar um método especifico (sem get ou set, ela é publica). Caso não fosse possivel acessa-la, poderiamos usar os métodos GETTERS e SETTERS, geralmente eles são usados para atributos do tipo PRIVATE. Outra maneira que seria possivel acessa-la seria (n1.getVida()).

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

**R:** A variavel “cont” está sendo incrementada de 1 a cada vez que o metodo “Nave” é instanciado para vida. Nesse código o valor maximo dela sera 3, pois foram instanciados 3 objetos. Como cont é static, a variavel está disponivel para toda classe.

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

**R:** Sim, pois na linha 10, temos um construtor parametrizado com STRING NOME.

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

**R:** Estado da classe são os valores atribuidos a um objeto. O comportamento da classe são os métodos.

estado -> carro; comportamento -> ligar carro. (void)

**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 da classe, é aquele construtor que tem o mesmo nome da classe. Ele pode ser implicito ou explicito (depende de cada caso), ele é usado para instanciar objetos. Sim, ele pode ser patametrizado.