Universidade da Amazônia

Curso: Análise e Des. de Sistemas / Ciência da Computação

Turma: ALC XXX 01 03 NXA

Disciplina: Programação Orientada a Objetos

Professor: Alan Souza

**LISTA DE EXERCÍCIO 3 - 2a Avaliação**

1)

Suponha os seguintes cenários com a programação orientada a objetos com Java:

I. A classe A possui um método privado e estático que pode ser acessado na classe B sem a necessidade de criar e usar um objeto.

II. A instrução “String email = JOptionPane.showInputDialog("Digite seu e-mail");” é um exemplo de acesso a um método estático sem erros.

III. O método toString() faz parte da classe Object.

Marque a alternativa correta:

a) Apenas as alternativas I e II estão corretas.
b) Apenas as alternativas I e III estão corretas.
c) Apenas as alternativas II e III estão corretas.
d) Apenas a alternativa II está correta.
e) Todas as alternativas estão corretas.

2)

Analise as asserções a seguir e a relação proposta entre elas:

I. A classe “public final class Boleto” não pode ser herdada por outra classe.

PORQUE

II. Está configurada como “final”.

A respeito dessas asserções, assinale a opção correta.

a) As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.
b) As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I.
c) A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
d) A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
e) As asserções I e II são proposições falsas.

3)

Analise o código a seguir:

```
try {
    double a = 10.0, b = 5.0;
    System.out.println(“Soma: ” + (a+b));
    System.out.println(“Subtração: ” + (a-b));
    System.out.println(“Divisão: ” + (a/b));
    System.out.println(“Multiplicação: ” + (a*b));
    System.out.println(“Tudo feito.”);
} catch(Exception e) {
    System.out.println(“Erro!”);
} finally {
    System.out.println(“Finalizando...”);
}
```

O que será impresso quando esse código for executado?

a) Soma: 15.0 / Subtração: 5.0 / Divisão: 2.0 / Multiplicação: 50.0 / Tudo feito. / Finalizando...
b) Soma: 15.0 / Subtração: 5.0 / Divisão: 2.0 / Multiplicação: 50.0 / Tudo feito.
c) Soma: 15.0 / Subtração: 5.0 / Divisão: 2.0 / Multiplicação: 50.0
d) Erro! / Finalizando...
e) Erro!

4)

Analise as asserções a seguir e a relação proposta entre elas:

I. O comando “public class A implements X, Y” gera erro enquanto os métodos declarados em X e Y não forem sobrescritos.

PORQUE

II. Muito provavelmente os métodos abstratos das classes X e Y precisam ser implementados na classe A.

A respeito dessas asserções, assinale a opção correta.

a) As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.
b) As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I.
c) A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
d) A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
e) As asserções I e II são proposições falsas.

5)

Sobre métodos/funções de classes e objetos em Java, analise as afirmativas abaixo e assinale a alternativa correta.

I - Podem receber e/ou retornar argumentos (valores).

II - A variável que receber o valor retornado pelo método deve ser do mesmo tipo do retorno do método.

III - O tipo de variável enviado para o método deve ser diferente do tipo de variável de recebimento do método.

a) As afirmativas I e II estão corretas.
b) Apenas a afirmativa I está correta.
c) Apenas a afirmativa II está correta.
d) Apenas a afirmativa III está correta.
e) As afirmativas I, II e III estão corretas.

6)

Observe as afirmações abaixo e marque a alternativa correta:

I - A classe Veículo pode herda da classe Carro.

II - A classe Professor pode herdar da classe Pessoa.

III - A classe Refrigerante pode herdar da classe Bebida

Está verdadeiro o que se afirma em:

a) Apenas I.
b) Apenas II.
c) Apenas III.
d) I e II apenas.
e) II e III apenas.

7)

Julgue os itens a seguir:

I. O construtor parametrizado recebe parâmetros que serão armazenados nos atributos da classe na hora da criação de um objeto.

II. O construtor não parametrizado deve ser sempre público.

III. Não se usa construtor para criar objetos.

Marque a alternativa correta:

a) I, II,II são verdadeiros.
b) Apenas I é verdadeiro.
c) Apenas II é verdadeiro.
d) Apenas III é verdadeiro.
e) Apenas II e III são verdadeiros.

8)

Em relação a herança em programação orientada a objetos:

I. Classes não podem herdar atributos e métodos de outra classe.

II. Em Java, não se pode herdar de duas classes ao mesmo tempo (herança múltipla).

III. Atributos e métodos configurados como “private” são herdáveis.

IV. Na fase de projeto: perguntar É-UM.

São verdadeiras as afirmações:

a) I, II, III, IV.
b) I e III apenas.
c) II e IV apenas.
d) I e IV apenas.
e) II, III e IV apenas.

9)

Avalie as asserções a seguir e a relação proposta entre elas.

I. Atributos com o modificador de acesso “private” precisam de métodos setters e getters públicos para serem acessados em outra classe.

PORQUE

II. Atributos com modificador de acesso “private” só podem ser acessados por classes que fazem parte do mesmo pacote.

A respeito dessas asserções, assinale a opção correta.

a) As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.
b) As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I.
c) A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
d) A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
e) As asserções I e II são proposições falsas.

10)

Analise os itens a seguir:

I. O encapsulamento não é uma das principais técnicas que define a programação orientada a objetos.

II. O reaproveitamento de código é uma das grandes vantagens da programação orientada a objetos. Muito disso se dá por causa do encapsulamento das classes.

III. Um assunto crucial na programação orientada a objetos é o chamado polimorfismo.

IV. A abstração consiste em um dos pontos mais importantes dentro de qualquer linguagem orientada a objetos.

São verdadeiras as afirmações:

a) I, II, III e IV.
b) I e II apenas.
c) II e III apenas.
d) III e IV apenas.
e) IV apenas.

11)

Explique como funcionam as classes, métodos e atributos declarados com “final”.

12)

Em relação ao polimorfismo na programação orientada a objetos com Java, explique como se dá a sobrecarga e a sobrescrita de métodos, citando um exemplo para cada um.

**Gabarito:**

1) C

2) A

3) A

4) C

5) A

6) E

7) B

8) C

9) C

10) D

11) R: Classe final: não pode ser herdada por outra classe.

Método final: não pode ser sobrescrito na classe filha.

Atributo final: não pode ter o seu valor alterado, ou seja, é uma constante.

12) R: **Sobrecarga**: ocorre quando duas ou mais estruturas semelhantes (método ou construtor) possuem variação na quantidade de elementos de entrada.

Exemplo 1 de sobrecarga:

```
public void metodo1(String s) { ... }
public void metodo1(double n, double m) { ... }
```

Exemplo 2 de sobrecarga: construtor padrão e construtor parametrizado:

```
public class Carro {
   private String modelo;
   public Carro() { }
   public Carro(String modelo) {
       this.modelo = modelo;
}
}
```

Criação do objeto:

```
Carro c1 = new Carro(“hatch”);
Carro c2 = new Carro();
```

**Sobrescrita**: ocorre quando existe herança entre duas classes e a subclasse sobrescreve o comportamento da super classe.

Exemplos: O método toString() da classe Object imprime o endereço na memória do objeto, mas pode ser sobrescrita na subclasse para imprimir seus atributos.

```
public class Transacao {
   // atributos...
   public void imprimirComprovante() {
       // imprimir os atributos de Transacao
   }
}
public class Transferencia extends Transacao {
   // atributos...
   @Override
   public void imprimirComprovante() {
       super.imprimirComprovante();
```

```
        // imprimir os atributos de Transferencia
    }
}
public class Pessoa extends Object {
    private String nome;
    private double altura;
    @Override
    public String toString() {
        return “Pessoa{nome: “+this.nome+”, altura: ”+this.altura+”}”;
    }
}
```