

Parte I - Conceitos teóricos



A programação Orientada a objetos (POO) é uma forma especial de programar, mais próximo de como expressaríamos as coisas na vida real do que outros tipos de programação. *Nesse contexto*, quanto aos conceitos e características da orientados a objeto, faça as questões de teóricas (múltipla escolha, objetiva, dissertativo, verdadeiro ou falso) com objetivo de mensurar aprendizagem do conteúdo do programa de ensino.

1) Na classe Conta, o método calcularBonificacao possui a seguinte assinatura:

```
public void calcularBonificacao ( )
```

Como você invoca este método?

- (A) Conta conta = new Conta(); conta.calcularBonificacao (8);
- (B) Conta conta = conta.calcularBonificacao (8 );
- (C) Conta (calcularBonificacao ( ));
- (D) Conta conta = new Conta(); conta.calcularBonificacao ( );

2) Os métodos em Java que não retornam valores devem possuir no parâmetro tipo-de-retorno a palavra?

- (A) void .
- (B) static.
- (C) main.
- (D) String args.
- (E) public.

3) Na classe Conta, o método **calcularBonificacao** possui a seguinte assinatura:

```
public double calcularBonificacao ( float pSalario ,  
                                     int pTotalDias, double pDesconto )
```

Quantos argumentos **calcularBonificacao** têm?

- (A) 0
- (B) 1
- (C) 2
- (D) 3

4) Na classe Conta, o método **calcularBonificacao** possui a seguinte assinatura:

```
public double calcularBonificacao ( float pSalario ,  
                                     int pTotalDias, String pAviso )
```

Qual é o tipo de retorno do método **calcularBonificacao**?

- (A) double
- (B) int
- (C) float
- (D) String

5) No contexto da Orientação a Objetos, uma entidade que combina estrutura de dados (atributos) e comportamento(método) é denominado?

- (A) Classe
- (B) Herança
- (C) objeto
- (D) Encapsulamento
- (E) Polimorfismo

- 6) A respeito dos conceitos fundamentais de programação orientada a objeto, julgue os itens seguintes designando (C) para os certos e (E) para os errados:
- (a) (C) (E) - Em programação orientada a objetos, as propriedades que definem a estrutura (atributo) e o comportamento (método) de um objeto são especificadas na classe da qual o objeto é instância e são válidas para todos os objetos dessa classe.
  - (b) (C) (E) Todo objeto tem um identificador único que varia com o tempo de vida do objeto.
  - (c) (C) (E) Enquanto a programação estruturada tem por característica desenvolver uma série de funções ou algoritmos para tratar determinado problema na perspectiva de um conjunto particular de dados, a programação orientada a objetos considera primeiramente os dados (atributos) para, então, identificar o comportamento (método) para atualizá-los ou consultá-los; cada objeto é responsável por executar uma série de tarefas
  - (d) (C) (E) os métodos podem apresentar uma série de características definidas a partir de palavras-chave utilizadas na sua definição. A palavra chave "private" indica que o método é acessível para somente dentro da classe.
  - (e) (C) (E) Uma classe é composta das regras que definem o objeto
  - (f) (C) (E) Um objeto possui duas características essenciais: estado que são as características (atributos) que definem o objeto e comportamento que é um conjunto de ações pré-definidas (denominada métodos).
  - (g) (C) (E) Uma Classe concreta é uma classe que pode possui atributos, métodos, construtores e que pode ser instanciada, ou seja, permite a criação de objetos a partir dela

- 7) A respeito dos conceitos fundamentais de programação orientada a objeto, julgue os itens seguintes designando (C) para os certos e (E) para os errados:
- (a) (C) (E) métodos é o código de programação que define o comportamento do objeto.
  - (b) (C) (E) O comportamento de um objeto é definido em sua respectiva classe, por meio da implementação de métodos que são executados quando tal objeto recebe uma mensagem (invocação).
  - (c) (C) (E) os métodos podem apresentar uma série de características definidas a partir de palavras-chave utilizadas na sua definição. A palavra chave "public" indica que o método é acessível para todas as classes do sistema.
  - (d) (C) (E) os métodos podem apresentar uma série de características definidas a partir de palavras-chave utilizadas na sua definição. A palavra-chave "private" especifica que o método é acessível para qualquer classe da aplicação.
  - (e) (C) (E) O modificador de acesso public, indica que a método pode ser utilizada por qualquer classe da aplicação
  - (f) (C) (E) os métodos *public* de uma classe são utilizados pelos clientes(programa) da classe para manipular dados armazenados em objetos dessa classe
  - (g) (C) (E) o método *private* de uma classe, esse método é privativo da classe que o contém, sendo seu uso permitido apenas dentro de uma mesma classe, e vedado o acesso a qualquer outra classe.
  - (h) (C) (E) A assinatura dos métodos de uma classe é composta por nome do método, tipo de retorno, lista de argumentos e modificador de acesso
  - (i) (C) (E) Todo objeto tem um identificador único durante todo tempo de vida do objeto.

- 8) No contexto da orientação a objetos, é correto afirmar sobre o conceito de objeto ?
- (A) uma rotina de programação contida em uma classe que pode ser chamada diversas vezes possibilitando assim reuso de código de programação.
  - (B) um conjunto de atributos primitivos tipados contido em uma classe.
  - (C) Um objeto é a instância de uma classe. Assim, objeto tem estado(atributos), comportamento(métodos) e identidade.
  - (D) um elemento de uma classe que representa uma operação (a implementação de uma operação).
  - (E) uma porção de código que resolve um problema muito específico, parte de um problema maior.
- 9) Marque a opção que possui uma assinatura ERRADA de um método em Java:
- (A) `private void imprimir(String s)`
  - (B) `public static int adicionar(int a, int b)`
  - (C) `protected double dividir(double i, double j)`
  - (D) `void exponenciacao(double x; double n)`
  - (E) `static int subtrair(double x, double y)`

- 10) Assinale a única alternativa CORRETA sobre modificadores de acesso:
- (A) Métodos e atributos public podem ser vistos por qualquer classe, menos por aquelas fora do pacote.
  - (B) Atributos default(pacote) não podem ser vistos por classes do mesmo pacote.
  - (C) Quando o desenvolvedor não coloca modificador de acesso em atributos de classes, por padrão eles serão public.
  - (D) Métodos protected podem ser vistos por classes filhas, mas só se elas estiverem no mesmo pacote que a classe do método protected.
  - (E) Métodos private só são visíveis dentro da classe onde estão.
- 11) Uma metodologia de desenvolvimento de sistemas é considerada Orientada a Objetos se ela orienta a construção de sistemas a partir do entendimento do mundo real como um conjunto de objetos que se comunicam entre si de forma coordenada. Dentre os principais conceitos relativos à Orientação a Objetos pode-se destacar (1) Métodos. De acordo com as aulas ministradas, responda aos itens abaixo:

(A) Defina o conceito de Método?

---

---

---

---

---

Parte II - Lendo e entendendo código - objetivo é compreender a estrutura sintaxe da linguagem

Para as questões 1 a 3, considere que a classe Aluno tenha dois métodos com as seguintes assinaturas; também temos um construtor padrão já codificado.

```
public class Aluno {  
    private final double VALOR_PIS = 0.65;  
    private int totalAlunos;  
    private double nota;  
    private String nome;  
  
    public Aluno() {  
        super();  
        // TODO Auto-generated constructor stub  
    }  
    public int getTotal(int pNumero){  
    public double calcularNota(double pNP1, double pNP2){  
}
```

- 1) Qual é o tipo de dado do atributo totalAlunos?
  - (A) float
  - (B) double
  - (C) String
  - (D) Int
- 2) Qual é o tipo de retorno do método calcularNota?
  - (A) String
  - (B) double
  - (C) int
  - (D) objeto

```
public class Aluno {  
    private final double VALOR_PIS = 0.65;  
    private int totalAlunos;  
    private double nota;  
    private String nome;  
  
    public Aluno() {  
        super();  
        // TODO Auto-generated constructor stub  
    }  
    public int getTotal(int pNumero){  
    public double calcularNota(double pNP1, double pNP2){  
}
```

- 3) Instancie um objeto da classe Aluno e invoque os seguintes métodos getTotal passando 12 como parâmetro e calcularNota passando com parâmetros 8 e 7.5; a partir de uma classe TesteAluno



4) A classe Java a seguir deve ser utilizada para responder a questões

```
public class Calculo {  
  
    public double soma(double pNumero1, double pNumero2) {  
        return pNumero1 + pNumero2;  
    }  
  
    public double soma(double pNumero1, double pNumero2, double pNumero3) {  
        return pNumero1 + pNumero2 + pNumero3;  
    }  
}
```

Para invocar os dois métodos **soma** da classe **Calculo** e mostrar na tela o valor de retorno, é correto utilizar?

- (A) Exclusivamente as instruções
- Calculo calculo = new Calculo();
  - System.out.println (calculo.soma(10, 20));
- (B) Exclusivamente a instrução
- System.out.println (Calculo.soma(10, 20, 50));
- (C) Exclusivamente as instruções
- Calculo calculo = new Calculo();
  - double r = calculo.soma(10, 20);
  - System.out.println(r);
- (D) Calculo calculo = Calculo.soma(10,20,30);
- System.out.println (calculo);
  - System.out.println (Calculo.soma(10, 20));
- (E) Calculo calculo = new Calculo();
- System.out.println (calculo.soma(10, 20, 30));
  - System.out.println (calculo.soma(10, 20));

- 5) Analise o algoritmo a seguir implementado em Java para responder a questão abaixo?

```
public class FolhaPagamento {  
  
    public double calcularCredito(double pValorInicial, double pValorBasico, int pFator) {  
        double calculo1 = pValorInicial + pValorBasico;  
        double calculo2 = calculo1 * (pFator + 5);  
        return calculo2;  
    }  
}
```

Se o método calcularCredito receber como parâmetros, respectivamente, os valores numéricos 10, 5, 2, ele retornará o valor:

- (A) 95
  - (B) Nenhum valor
  - (C) 75
  - (D) 125
  - (E) 105
- 6) Analise o algoritmo a seguir implementado em Java para responder a questão abaixo?

```
public void calcularImposto(double pValor, double pPorcentagem) {  
    valorImposto = pValor * pPorcentagem ;  
}
```

Se o método calcularImposto receber como parâmetros, respectivamente, os valores numéricos 500, 0.10, ele retornará o valor:

- (A) 50
- (B) 85
- (C) 75
- (D) Nenhum valor
- (E) 105