Nota: 4.00

Início da Prova

---

1. [CODATA-PB/2013] Na orientação a objetos, a sobrecarga é utilizada por meio do conceito de:

a) polimorfismo.
b) herança.
c) encapsulamento.
d) agregação.
e) abstração.

Explicações:

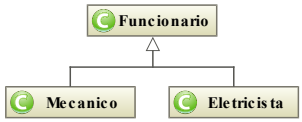
- Sobrecarga é considerado um tipo de polimorfismo, uma vez que a execução de um método (pelo nome dele) vai ter comportamento diferente dependendo dos parâmetros passados. Também conhecido como polimorfismo ad-hoc.

Ações:

- Dúvidas e Discussões

---

2. [IBGE/2016] A figura a seguir representa o diagrama de classes de parte do sistema SOMANUT. Na classe Funcionario, foi implementado o método abstrato calcularSalario.



Para que a classe Mecanico possa ser instanciada, é necessário que a(s) classe(s):

a) Mecanico e Eletricista redefinam o método calcularSalario;
b) Mecanico redefina o método calcularSalario;
c) Funcionario tenha definido um atributo salario que seja protegido;
d) Funcionario tenha definido um atributo salario que seja público;
e) Funcionario possua outros métodos concretos.

Ações:

- Dúvidas e Discussões

---

3. [ENADE/2014] O código a seguir mostra um programa escrito na linguagem de programação Java.

```
class Java {
  static void trocaB(String a, String b) {
    String tmp = a;
    a = b;
    b = tmp;
  }

  static void trocaC(int array[], String a) {
    for (int x=0; x<array.length; x++) {
      array[x] = array[x] * Integer.valueOf(a);
    }
  }

  public static void main(String[] args) {
    int array[] = { 1, 2, 3, 4, 5};
    String a = "2", b = "5";
    trocaB(a, b);
    trocaC(array, a);
    System.out.print(a + " " + b + " ");
    for (int x=0; x<array.length; x++) {
      System.out.print(array[x] + " ");
    }
  }
}
```

Após executado o código, o valor impresso na saída padrão do usuário será:

a) 5 2 1 2 3 4 5.
b) 2 5 2 4 6 8 10.
c) 5 2 5 10 15 20 25.
d) 5 2 2 4 6 8 10.
e) 2 5 1 2 3 4 5.

Explicações:

- No caso do método `trocaB`, os valores de `a` e `b` não são trocados porque, apesar de eles serem referências para objetos, dentro do método não estamos mudando nenhum atributo dos objetos. O que estamos fazendo, dentro do método e no caso do parâmetro `a`, é mudar o endereço inicial que estava sendo armazenado no parâmetro para apontar para outro endereço. Note que o conteúdo que estava no endereço original nunca foi mudado.
- Para piorar, neste caso específico das strings, lembre-se que em Java toda string é imutável, ou seja, não é possível mudar seu conteúdo. O que acontece quando "mudamos" uma string é que um novo objeto é criado e a variável passa a apontar para a nova referência. Resumindo: é impossível mudar uma String que foi passada como parâmetro. Mas, apenas para evitar mal entendimento, note que teríamos o mesmo comportamento para parâmetros de qualquer classe no caso da implementação do método `trocaB`. Essa observação do parâmetro ser string, é só um detalhe.
- Já no caso do método `trocaC`, o que está sendo passado é uma referência para um vetor. E note que, dentro do método, estamos mudando os valores do vetor, ou seja, o conteúdo da memória do vetor original. Como estamos mudando o conteúdo da referência, o valor original (do vetor) também mudará.
- Essa questão é uma das mais difíceis para quem não tem domínio completo de como as referências funcionam (em qualquer linguagem, e não só em Java). Se você não consegue entender 100% do que está acontecendo no código, é importantíssimo que você busque mais informações e se esforce para entender, pois isso significa que você tem uma visão errada ou incompleta de como as referências funcionam.

Ações:

- Dúvidas e Discussões

---

4. [ENADE/2008] Uma pizzaria fez uma ampliação de suas instalações e o gerente aproveitou para melhorar o sistema informatizado, que era limitado e não atendia a todas as funções necessárias. O gerente, então, contratou uma empresa para ampliar o software. No desenvolvimento do novo sistema, a empresa aproveitou partes do sistema antigo e estendeu os componentes de maneira a usar código validado, acrescentando as novas funções solicitadas.

Que conceito de orientação a objeto está descrito na situação hipotética acima?

a) Abstração
b) Herança
c) Sobrecarga
d) Sobreposição
e) Mensagem

Explicações:

- Herança é o conceito de orientação a objetos que permite pegar uma classe (código) existente e estendê-la com novas funcionalidades e/ou mudar a implementação das funcionalidades existentes.

Ações:

- Dúvidas e Discussões

---

5. [PosComp/2007] Analise as seguintes afirmativas.

I. Encapsulamento é o compartilhamento de atributos e métodos entre classes com base em um relacionamento hierárquico.
II. Polimorfismo é capacidade de uma operação atuar de modos diversos em classes diferentes.
III. Herança consiste no processo de ocultação dos detalhes internos de implementação de um objeto.
IV. Sobreposição é a redefinição das funções de um método herdado. Os métodos apresentam assinaturas iguais.
V. Em JAVA, todos os métodos numa classe abstrata devem ser declarados como abstratos.

A partir da análise, pode-se concluir que

a) apenas as afirmativas I, III e V estão corretas.
b) apenas as afirmativas II e IV estão corretas.
c) apenas a afirmativa IV está correta.
d) apenas as afirmativas I, IV e V estão corretas.
e) todas as afirmativas são falsas.

Explicações:

- *Afirmativa I:* o conceito mencionado é o de *herança*.
- *Afirmativa III:* o conceito mencionado é o de *encapsulamento*.
- *Afirmativa V:* uma classe abstrata pode conter sim métodos concretos. Mas uma classe que possui pelo menos um método abstrato, deve obrigatoriamente ser declarada como abstrata.

Ações:

- Dúvidas e Discussões

---

6. [TCE-PR/2011] Em relação à Programação Orientada a Objetos, é INCORRETO afirmar:

a) No contexto da herança, uma instância da subclasse é, também, uma instância da superclasse.

b) Uma agregação representa um todo que é composto de várias partes e constitui um relacionamento de contenção; se qualquer uma das partes for destruída, as demais partes também o serão.

c) Polimorfismo pode ser entendido como um conceito complementar ao de herança. Assim, no polimorfismo é possível enviar a mesma mensagem a diferentes objetos e cada objeto responder da maneira mais apropriada para sua classe.

d) Interfaces são como as classes abstratas, mas nelas não é possível implementar nenhum método, apenas declarar suas assinaturas; uma classe ao implementar uma interface deverá escrever todos os seus métodos.

e) A aplicação do polimorfismo por inclusão requer que o método polimórfico seja declarado e/ou implementado na classe ancestral para possibilitar sua redefinição nas classes descendentes.

Ações:

- Dúvidas e Discussões

---

7. [TAP/2017] Observe o seguinte código em Java:

```
class Prova {
  static int v1;
  int v2;

  Prova(int p) { v2 = p; }

  void troca() { v1 = v2; }

  public static void main(String[] args) {
    Prova a = new Prova(5);
    Prova b = new Prova(2);
    a.troca();
    System.out.print(a.v1 + " " + a.v2 + " ");
    System.out.print(b.v1 + " " + b.v2);
  }
}
```

Assinale a alternativa correta:

a) O código compila e a saída é `"5 5 0 2"`.

b) O código compila e a saída é `"5 5 5 2"`.

c) O código compila e a saída é `"5 5 5 5"`.

d) O código compila e a saída é `"5 5 2 2"`.

e) O código não compila.

Explicações:

- Como o atributo `v1` é estático, ao mudar o seu valor a partir de um objeto, este valor muda para todos os outros objetos.
- Já o atributo `v2`, é de instância, e terá posições diferentes na memória para cada objeto.

Ações:

- Dúvidas e Discussões

---

8. [TAP/2018] Os operadores `*`, `/`, e `%` são chamados de operadores multiplicativos. Eles possuem a mesma precedência e são executados da esquerda para a direita (*left-associative*). Sabendo disso, considere o programa a seguir (Fonte: Bloch e Gafter, 2005):

```
public class Clock {
  public static void main(String[] args) {
    int minutes = 0;
    for (int ms = 0; ms < 60*60*1000; ms++)
      if (ms % 60*1000 == 0)
        minutes++;
    System.out.println(minutes);
  }
}
```

Este programa imprime:

a) 60000

b) 60

c) 0

d) 3600

e) 3600000

Explicações:

- A expressão `ms % 60*1000 == 0` será executado da seguinte forma: `((ms % 60) * 1000) == 0`.
- Considerando apenas a expressão aritmética (`ms % 60 * 1000`), esta será executada da esquerda para a direita, de acordo com o enunciado. Note que na questão espaços foram colocados ao redor do operador `%`, mas a presença ou não de espaços não altera a execução das expressões.

- Por fim, o operador de igualdade == será executado, pois ele possui menor precedência.
- Na dúvida, use sempre parênteses.

Mais informações:

- Java Language Specification - Multiplicative Operators

Ações:

- Dúvidas e Discussões

---

9. [TRE-RN/2011] Considere os seguintes fragmentos de código Java:

I
```
int sum = 7; if ( sum > 20 ) { System.out.print("ganhou");}else{
System.out.print("perdeu");} System.out.println(" o bônus.");
```

II
```
int sum = 21; if( sum!=20) System.out.print("ganhou");else
System.out.print("perdeu"); System.out.println(" o bônus.");
```

O resultado da execução dos fragmentos em I e II será, respectivamente,

a) perdeu o bônus e ganhou
b) perdeu e perdeu
c) ganhou e ganhou
d) perdeu o bônus e ganhou o bônus
e) perdeu e ganhou o bônus

Ações:

- Dúvidas e Discussões

---

10. [TAP/2017] Observe o seguinte código em Java (cada classe em seu arquivo):

```
public abstract class A {
  protected int i = 1;
  public abstract void print();
}
```

```
public class B extends A {
  public void print(int i) {
    System.out.println(i);
  }
  public static void main(String[] args) {
    B b = new B();
    b.print(42);
  }
}
```

Assinale a alternativa correta:

a) Classe A não compila (e nem a classe B).
b) O código compila e a saída é "0" (zero).
c) O código compila e a saída é "1" (um).
d) O código compila e a saída é "42".
e) Classe A compila, mas a Classe B não.

Explicações:

- A classe `B` não compila, pois o método `print` (sem parâmetros) da classe `A` (que é abstrato) não é sobreposto.
- Note que um método `print` foi implementado, mas este possui um parâmetro, o que o torna diferente do método com mesmo nome mas sem parâmetros (sobrecarga).

Ações:

- Dúvidas e Discussões

Clique no botão abaixo para fazer a prova novamente.

Resetar Respostas e Fazer a Prova Novamente