

Questão 6 (ID 6357)

A classe Funcionário possui três métodos que podem ser aplicados aos seus objetos: cadastrar, alterar e excluir. Considere que Gerente e Vendedor são classes derivadas da classe Funcionário e, por isso, herdam os métodos cadastrar, alterar e excluir, mas estes métodos são sobrescritos na classe Gerente e Vendedor com implementações bastante distintas, em função dos dados associados a cada um deles.

Partindo do exposto, assinale a opção que indica o conceito apresentado no texto.

- a) Encapsulamento.
- b) Reusabilidade.
- c) Abstração.
- d) Hereditariedade.
- e) Polimorfismo.

Questão 10 (ID 6377)

Em relação ao tema Programação Orientada a Objetos, analise as seguintes afirmativas:

I. Métodos são similares a procedimentos e funções e consistem nas descrições das operações que um objeto executa quando recebe uma mensagem.

II. Um atributo consiste em um dado ou informação de estado para o qual cada objeto de uma classe tem seu próprio valor.

III. Instância de Classe, diferentemente de objeto, é uma ocorrência específica de uma classe.

Está correto apenas o que se afirma em:

- a) I e III.
- b) II.
- c) III.
- d) I.
- e) I e II.

Questão 13 (ID 6372)

A característica que usamos para poder implementar os conhecimentos ou ações de uma classe, sem se preocupar como isto é feito, é denominada de:

- a) Herança.
- b) Encapsulamento.
- c) Classificação.
- d) Composição.
- e) Persistência.

Questão 14 (ID 6381)

Assinale a opção que define o termo Orientação a Objetos.

- a) É um paradigma de programação que descreve a computação como ações, enunciados ou comandos que mudam o estado (variáveis) de um programa.
- b) É uma forma de programação de computadores que preconiza que todos os programas possíveis podem ser reduzidos a apenas três estruturas: sequência, decisão e iteração.
- c) É um paradigma de programação que se refere ao uso de restrições na construção de relações entre variáveis.
- d) É um paradigma de programação que trata a computação como uma avaliação de funções matemáticas e que evita estados ou dados mutáveis.
- e) É um paradigma para o desenvolvimento de *software* que se baseia na utilização de componentes individuais (objetos) que colaboram para construir sistemas mais complexos.

Questão 15 (ID 6380)

Em relação ao tema Programação Orientada a Objetos, analise as seguintes afirmativas.

I. Polimorfismo é a propriedade que permite que a mesma mensagem seja enviada a diferentes objetos e que cada objeto execute a operação que é apropriada à sua classe.

II. A herança é permitida através da definição de uma hierarquia de classes, isto é, uma “árvore” de classes onde cada uma possui zero ou mais subclasses.

III. Objeto é elemento do mundo real (natureza). Sinônimo de instância de classe.

Está correto o que se afirma em:

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

Questão 17 (ID 6376)

Em relação ao tema Programação Orientada a Objetos, é **incorreto** afirmar que:

- a) Polimorfismo é a capacidade de um objeto poder ser referenciado de várias formas.
- b) Orientação a objetos é uma maneira de programar que ajuda na organização, mas não consegue resolver alguns dos problemas enfrentados pela programação procedural.
- c) A partir do momento em que você declara um construtor, o construtor default não é mais fornecido.
- d) Quando você não declara nenhum construtor na sua classe, o Java cria um para você. Esse construtor é o construtor default; ele não recebe nenhum argumento e o corpo dele é vazio.
- e) Orientação a objetos, além de concentrar as responsabilidades nos pontos certos, flexibiliza a aplicação, encapsulando a lógica de negócios.

Questão 18 (ID 6373)

Assinale a opção que indica o que permite a existência de vários métodos de mesmo nome, porém com assinaturas levemente diferentes.

- a) Composição.
- b) Herança.
- c) Sobrecarga.
- d) Classificação.
- e) Persistência.

Questão 21 (ID 6371)

Assinale a opção que indica qual é a propriedade que serve para criar classes que incorporam propriedades e métodos de outras classes, considerando a possibilidade de construção uma classe a partir de outra sem ter que reescrevê-la.

- a) Encapsulamento.
- b) Mensagem.
- c) Herança.
- d) Abstração.
- e) Polimorfismo.

Questão 1 (ID 53480)

O comportamento das classes seja modificado através da herança, no artigo usa-se o exemplo hipotético de um aplicativo gráfico. Uma parte de código qualquer, desse aplicativo hipotético, terá um apontador para um objeto que será responsável por desenhar um retângulo e outro apontador para um objeto capaz de renderizar texto, ambos os objetos renderizadores terão a função *draw()* que deve se comportar diferente para cada caso, um, desenhando retângulos e outro texto.



Com base na figura apresentada, é correto afirmar que o professor universitário pode ser considerado:

Com base na figura apresentada, é correto afirmar que o professor universitário pode ser considerado:

- a) Um método.
- b) Um polimorfismo.
- c) Uma herança.
- d) Uma abstração.
- e) Um encapsulamento.

Questão 13 (ID 53415)

Considere que um programa orientado a objeto possui 5 classes: Máquina, Motor, MotorExplosão, MotorVapor e Gerador. MotorExplosão e MotorVapor são especializações de Motor. Motor e Gerador são especializações de Máquina. Todas as classes respondem a uma mensagem chamada "calcularPotencia", sem argumentos, que calcula e retorna um número real que indica potência do objeto, em watts, de acordo com os valores de alguns atributos, com um algoritmo diferente em cada classe. O exemplo acima caracteriza a capacidade de enviar a mesma mensagem para vários objetos e que cada objeto responda a essa mensagem de acordo com sua classe. Tal característica é conhecida como:

- a) Independência de Dados.
- b) Tratamento de Exceção.
- c) Herança Múltipla.
- d) Polimorfismo.
- e) Refatoração.

Questão 27 (ID 53487)

Paradigma de programação relativamente recente, que se baseia fundamentalmente no conceito de Objetos. É uma forma de desenvolvimento de sistemas de software que o trata como um conjunto de componentes que interagem entre si para resolver um problema. Esses componentes são denominados Objetos. Os objetos são os "blocos de construção" de software na OO. Para uma linguagem de programação ser considerada no paradigma Orientada a Objetos, é necessário 4 pilares importantes. Quais são eles? Assinale a opção correta.

- a) Interface, Classe, Objeto, Atributo.
- b) Estrutura de dados, Interface, Pacotes, Polimorfismo.
- c) Polimorfismo, Instância, Objeto, Classe.
- d) Abstração, Encapsulamento, Herança, Polimorfismo.
- e) Mensagem, Interface, Pacotes, Atributo.

Questão 18 (ID 53417)

De acordo com os conceitos de Orientação a Objetos, o trecho de código abaixo é um exemplo clássico de:

```
public class Pessoa {
    private String nome;
    private String cpf;
    //...
    public Pessoa(String nome, String cpf){
        this.setNome(nome);
        this.setCpf(cpf);
    }
    //...
}

public class Professor extends Pessoa {
    private String codFuncional;
    //...
    public Professor(String nome, String cpf,
        String codFuncional) {
        super(nome, cpf);
        this.setCodFuncional(codFuncional);
    }
    //...
}
```

- a) Herança.
- b) Enumeração.
- c) Polimorfismo.
- d) Encapsulamento.
- e) Reescrita de classes.

Analisar o seguinte código Java e assinalar a opção que representa uma afirmação **incorreta**.

```
1 public class Data {  
2     private int dia;  
3     private int mes;  
4     private int ano;  
5  
6     public Data(dia, mes, ano){  
7         this.dia = dia;  
8         this.mes = mes;  
9         this.ano = ano;  
10    }  
11  
12    public static void main(String args[]){  
13        data hoje = new Data(2014, 05, 17);  
14    }  
15 }
```

a) Os atributos da classe Data são privados para evitar que entidades externas modifiquem os seus valores. Esta classe, na perspectiva ideal, deveria implementar os métodos get e set para cada um destes atributos.

b) A linha 13 contém um erro de programação, pois, dada a implementação do construtor de Data, espera-se que o primeiro argumento seja referente ao dia. No entanto, este erro não será detectado em tempo de compilação, pois os tipos são compatíveis.

c) O método main, nesta classe, tem apenas a função de instanciar um objeto do tipo Data, e pode ser utilizado para validar se o construtor está funcionando da maneira esperada.

d) Nas linhas 7, 8 e 9, a referência ao "this" indica que o valor dos parâmetros recebidos no método devem ser atribuídos aos atributos do objeto.

e) A classe Data deveria ser declarada como uma classe privada ao invés de pública, pois todos os seus atributos contêm o modificador private.

Assinale qual é o resultado apresentado após a execução do seguinte código em linguagem de programação Java.

```
public class JavaApplication3 {  
    public static void main(String[] args) {  
        int x=1,y=2;  
        String z = "3";  
        System.out.println( x + y + z );  
    }  
}
```

- a) 33.
- b) 6.
- c) 000.
- d) 123.
- e) 0.

Considere o código abaixo.

```
public static void main(String[] args) {  
    int[] idades = new int[10];  
    for (int i = 0; i < 10; i++) {  
        idades[i] = i * 10;  
    }  
    for (int i = 0; i < 10; i++) {  
        System.out.println(idades[i]);  
    }  
}
```

Assinale a opção que apresenta qual o resultado será impresso por meio do código acima.

- a) 0; 10; 20; 30; 40; 50; 60; 70; 80; 90; 100.
- b) 0; 10; 20; 30; 40; 50; 60; 70; 80; 90.
- c) Erro pois o elemento i não pode ser encontrado.
- d) 0; 10; 30; 40; 50; 60; 70; 80; 90; 100.
- e) Erro pois a declaração do *array* está errada.

Qual a saída do programa abaixo?

```
public class testaClasste {  
    public static void main(String[] args) {  
        int x=2,y=3,z=4;  
        x += y;  
        y = ++x;  
        z = ++y;  
        System.out.print( x );  
        System.out.print( y );  
        System.out.print( z );  
    }  
}
```


Qual a saída do programa abaixo?

```
public class testaArray {  
    public static void main(String[] args) {  
        int x[] = new int[3];  
        x[0]=9;  
        x[1]=8;  
        x[2]=7;  
        for(int a:x) {  
            System.out.print(a);  
        }  
    }  
}
```

Dependendo das necessidades encontradas optamos por um array ou um arrayList. A escolha depende do escopo do problema.

Analise as afirmativas a seguir.

I. Um *array* tem um tamanho fixo desde que o objeto é criado. Esse tamanho é imutável. Se definirmos um tamanho máximo a ele, independente se todos os elementos estão ocupadas ou não, o seu tamanho permanece o mesmo.

II. O tamanho de um arrayList é definido pela quantidade de componentes que ele possuir. Podemos criá-lo com zero posições ou ainda adicionar itens conforme a necessidade, refletindo no aumento de seu tamanho.

III. Um *array* pode receber mais argumentos do que foi definido no início, conforme o arrayList, basta que importe uma biblioteca específica que altera a sua quantidade inicial.

IV. Um arrayList tem a mesma funcionalidade do *array*, exceto pelo fato de o *array* possuir tamanho dinâmico.

Está correto apenas o que se afirma em:

a) I, II e III.

b) IV.

c) II e III.

d) I e II.

e) II, III e IV.