

****

**Exercícios - Java Básico**

**Módulos 4, 5, 6 e 7**

1. Com relação à programação orientada a objetos (POO), avalie a afirmativa: o uso de objetos em POO facilita o desenvolvimento de*software*, pois os objetos são divididos em propriedades e métodos, que representam, respectivamente, suas características e funções.
2. Correta
3. Errada
4. Dentro do paradigma de programação orientada a objetos (POO), há um mecanismo pelo qual uma classe pode estender outra classe, aproveitando seus métodos e atributos. Assinale a alternativa que apresenta o nome desse mecanismo.
5. Herança
6. Mensagem
7. Encapsulamento
8. Polimorfismo
9. Subclasse
10. Baseados nos conceitos obtidos sobre programação orientada a objetos, podemos inferir que o polimorfismo é:
11. a multiplicidade de atributos de determinada classe.
12. a propriedade de um diagrama de classes ter múltiplas classes possuidoras de atributos.
13. a habilidade de um atributo ou variável poder identificar instâncias de classes com atributos dependentes.
14. a propriedade de uma instrução poder apontar para múltiplos objetos de uma mesma classe sem implicações de desempenho.
15. a habilidade pela qual uma única operação ou nome de atributo pode ser definido em mais de uma classe e assumir implementações diferentes em cada uma dessas classes.
16. Baseados nos conceitos obtidos sobre programação orientada a objetos, inferimos que o conceito de encapsulamento é utilizado quando:
17. os objetos de uma classe devem ser armazenados em tabelas de um gerenciador de banco de dados relacional.
18. todos os atributos de uma classe devem possuir o mesmo tipo de dados.
19. as classes devem possuir apenas atributos, sem nenhum método próprio.
20. detalhes da implementação de uma classe são ocultados das demais classes.
21. os objetos de uma classe devem ser armazenados diretamente em objetos de um gerenciador de banco de dados orientado a objetos.
22. Baseado nos conceitos obtidos sobre programação orientada a objetos, podemos inferir que o encapsulamento:
23. É a base de toda a abordagem dessa metodologia de programação e diz-se que um dado está encapsulado quando envolvido por código de forma que só é visível na rotina onde foi criado; o mesmo acontece com uma rotina, que sendo encapsulada, suas operações internas são invisíveis às outras rotinas.
24. Pode ser entendido como sendo um conjunto de instâncias criadas a partir de um outro conjunto de instâncias com características semelhantes.
25. É definido como sendo uma técnica que permite a um código possuir "vários comportamentos" ou produzir "vários comportamentos".
26. Possibilita a criação de uma nova classe de modo que essa classe (denominada subclasse, classe-filha ou classe derivada) herda todas as características da classe-mãe (denominada superclasse, classe base ou classe primitiva); podendo, ainda, a classe-filha possuir propriedades e métodos próprios.
27. É considerado como a habilidade de modelar características do mundo real do problema que o programador esteja tentando resolver.
28. No paradigma de programação orientada a objetos, a herança facilita:
29. A leitura e a escrita de código pelo compartilhamento de nomes dos métodos.
30. A reutilização e a modificação dos módulos de códigos existentes.
31. Esconder e proteger detalhes da implementação.
32. A definição de classes abstratas
33. A composição e interação entre os objetos.
34. Indique a palavra chave utilizada para declaração de herança em java:
35. heritage
36. extends
37. super
38. continue
39. Indique a palavra chave utilizada para acessar os métodos e atributos de uma classe que está sendo herdada.
40. this
41. implements
42. transient
43. super
44. Qual palavra chave utilizamos para criarmos uma constante em Java?
45. null
46. transient
47. final
48. abstract
49. Analise o trecho do código abaixo e indique se ele está correto ou não.

public final class MyClass {}

public class MyOtherClass extends MyClass {}

1. Correto
2. Errado

Res:.Uma classe final não pode ser errada por uma outra classe

1. Em relação a classes abstratas, indique a afirmativa correta.
2. Classes abstratas não podem ser instanciadas.
3. Para criarmos uma classe ou método abstrato usamos a palavra-chave static.
4. Uma classe abstrata poderá ser instanciada.
5. Uma classe abstrata precisa obrigatoriamente definir um método abstrato.
6. Em relação a interfaces, indique a afirmativa correta.
7. Uma interface pode conter vários construtores.
8. Um interface filha pode estender várias interfaces pai
9. Uma interface pode implementar várias interfaces
10. Uma interface pode ter o modificador private ou protected.
11. Analise o texto e identifique o modificador de acesso dentre as alternativas abaixo:

Este modificador de acesso torna o membro acessível às classes do mesmo pacote ou através de herança. Os seus membros herdados não são acessíveis a outras classes fora do pacote em que foram declarados.

1. Private
2. Public
3. Default
4. Protected
5. Analise o código abaixo e assinale a alternativa correta:

**public** **class** Carro {

**private** String cor(){

**return** "Azul";

}

}

**public** **class** Oficina {

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

Carro c = **new** Carro();

System.***out***.println("A cor do carro é: " + c.cor());

}

}

1. A saída será “A cor do carro é: Azul”;
2. Erro de compilação.
3. Erro de execução.
4. Nenhuma das alternativas acima.

Res:. Tinha que ser da classe alterar a classe oficina para extendes

1. Acerca do modificador de acesso *final*, analise as afirmativas abaixo e identifique a **incorreta**:
2. Uma vez atribuído valor, variáveis *final* não podem ter seus valores alterados, ou seja, são constantes.
3. Uma classe *final* não pode ser herdada
4. Métodos *final* podem ser sobrescritos.
5. Todas as alternativas acima estão incorretas.
6. Sobre o modificador de acesso *static*, analise as afirmativas abaixo e identifique aquela que está correta:
7. Este modificador de acesso só pode ser utilizado em variáveis;
8. Um atributo estático é compartilhado por todas as instâncias de uma classe;
9. A palavra chave *this* pode ser utilizada dentro de métodos estáticos para referenciar a instância sendo executada;
10. Esse modificador de acesso só poderá ser utilizado em métodos;

Res: ele virara constantes

1. Analise o texto abaixo e indique se a afirmativa está correta:

No modificador de acesso *default* (padrão) a classe e/ou seus membros são acessíveis somente por classes do mesmo pacote. Na sua declaração não é definido nenhum tipo de modificador.

1. Correta
2. Errada
3. Com o intuito de iniciar o desenvolvimento de um sistema de controle acadêmico, você foi solicitado implementar algumas classes básicas. Em Java, crie as seguintes classes e aplique o conhecimento adquirido acerca dos fundamentos em programação orientada a objetos:

* Pessoa (possui os atributos nome, CPF e idade);
* Aluno (possui o atributo matrícula e estende a classe pessoa);
* Professor (possui o atributo disciplina e estende a classe pessoa);
* Curso (possui os atributos código e nome);
* Turma (possui o atributo quantidade de alunos e extendes de curso);

Public class Pessoa

{

Private String nome;

Private int cpf;

Private String idade;

}

Get e set

Public class Aluno extends Pessoa

{

Private int matricula;

}

Public class Professor extends Pessoa

{

Private String disciplina;

}

Public class Curso extends Professor

{

Private int codigo;

Private String nome;

}

Public class Turma extends Curso

{

Private int quantidade\_alunos;

}

