

Data: 19/01/2026

Reposição P2

1. Qual palavra-chave é utilizada em Java para que uma classe herde de outra?
a) implements
b) inherits
c) extends
d) Super
e) instance

2. Java suporta herança múltipla de classes?
a) Sim, separando as classes por vírgula.
b) Não, uma classe pode ter apenas uma superclasse direta.
c) Sim, mas apenas se as classes forem abstratas.
d) Não, Java não suporta herança de forma alguma.
e) Sim, utilizando a palavra-chave virtual.

3. O que acontece se um método for marcado com @Override mas não corresponder a nenhum método da superclasse?

- a) O código compila, mas o método funciona como um novo método.
b) O compilador gera um erro.
c) Ocurre uma exceção em tempo de execução.
d) O método torna-se abstrato automaticamente.
e) O método é ignorado pelo sistema.

4. A palavra-chave super é usada para:

- a) Instanciar um objeto da classe atual.
b) Declarar uma superclasse.
c) Acessar membros (métodos/construtores) da superclasse imediata.
d) Definir uma classe que não pode ser estendida.
e) Transformar uma variável em constante.

5. Qual a principal característica de uma Classe Abstrata?

- a) Não pode ter métodos concretos (com corpo).
b) Todos os seus métodos devem ser estáticos.
c) Não pode ser instanciada diretamente (não se pode fazer new).
d) Não pode ter atributos.
e) É obrigada a implementar uma interface.

6. Uma classe que implementa uma Interface deve:
a) Objetivamente ser abstrata.
b) Fornecer implementação para todos os métodos abstratos da interface ou ser declarada abstrata.
c) Estender a interface com extends.
d) Ter apenas métodos privados.
e) Declarar seus métodos como protected.

7. Sobre atributos (variáveis) declarados dentro de uma Interface, eles são implicitamente:
a) private e final.
b) public, static e final (constantes).
c) protected e abstract.
d) private e static.
e) Apenas public.

8. O que é Polimorfismo em Java?
a) A capacidade de uma classe ter vários métodos com o mesmo nome (sobrecarga).
b) A capacidade de uma variável de referência de uma superclasse apontar para objetos de subclasses diferentes.
c) A capacidade de converter tipos primitivos entre si.
d) O uso de múltiplas interfaces em uma classe.
e) A criação de cópias exatas de um objeto.

9. Se temos Animal a = new Cachorro(); e chamamos a.emitirSom(), qual método será executado (considerando que Cachorro herda de Animal e sobrescreve o método emitirSom())?
a) O método da classe Animal.
b) O método da classe Cachorro.
c) Nenhum, ocorrerá erro de compilação.
d) Ambos os métodos sequencialmente.
e) O método da classe Object.

10. Qual modificador impede que uma classe seja herdada (estendida)?
a) static
b) abstract
c) final
d) private
e) protected

11. Qual é o objetivo de uma Classe Adaptadora (Adapter)?
a) Criar uma classe singleton.
b) Permitir que classes com interfaces incompatíveis trabalhem juntas.
c) Otimizar o uso de memória em heranças profundas.
d) Adaptar o código Java para rodar em C++.
e) Gerenciar conexões de banco de dados.

12. Em Java, uma classe pode implementar quantas interfaces?
a) Apenas uma.
b) Apenas duas.
c) Quantas forem necessárias (múltiplas).
d) Nenhuma, interfaces são apenas para design.
e) Até 10 interfaces.

13. A conversão explícita de uma referência de superclasse para subclasse (ex: (Cachorro) animal) é chamada de:
- a) Upcasting.
 - b) Downcasting.
 - c) Broadcasting.
 - d) Overloading.
 - e) Boxing.

C

14. O que ocorre no seguinte código se animal NÃO for um Cachorro?

Cachorro c = (Cachorro) animal;

- a) O objeto é convertido forçadamente.
- b) O valor null é atribuído.
- c) Lança ClassCastException em tempo de execução.
- d) Erro de compilação.
- e) O programa fecha silenciosamente.

C

15. Qual operador verifica se um objeto é uma instância de uma determinada classe ou interface?

- a) isA
- b) typeof
- c) instanceof
- d) checkType
- e) as:

C

16. Métodos abstratos podem existir em:

- a) Qualquer classe.
- b) Apenas em Interfaces.
- c) Apenas em Classes Abstratas e Interfaces.
- d) Apenas em Classes Finais.
- e) Apenas em Classes Estáticas.

C

17. O que é "Assinatura de Método"?

- a) O nome do método e sua lista de parâmetros (tipos e ordem).
- b) Apenas o nome do método.
- c) O nome, parâmetros e tipo de retorno.
- d) O modificador de acesso e o nome.
- e) O corpo do método.

X

18. Qual a visibilidade do modificador protected?

- a) Apenas na própria classe.
- b) Apenas no próprio pacote.
- c) Na própria classe, no mesmo pacote e em subclasses (mesmo em pacotes diferentes).
- d) Em todo o projeto (universo).
- e) Apenas em classes abstratas.

C

19. Uma classe abstrata pode ter construtor?

- a) Não, pois não pode ser instanciada.
- b) Sim, para ser invocado pelas subclasses via super().
- c) Sim, mas ele deve ser privado.
- d) Não, construtores são exclusivos de classes concretas.
- e) Sim, mas apenas se não tiver argumentos.

C

20. A interface funcional caracteriza-se por:

- a) Não ter nenhum método.
- b) Ter exatamente um método abstrato.
- c) Ter apenas métodos default.
- d) Ser usada apenas para GUI.
- e) Não poder ser implementada por classes.

C