

## Questões sobre Herança e Encapsulamento em Java

---

### Questões sobre Java (AWT, Swing e JavaFX)

1. Em Java, qual palavra-chave é usada para indicar que uma classe herda de outra?
  - a) ☐ inherit
  - b) ☐ super
  - c) ☒ extends
  - d) ☐ implements
  
2. Qual palavra-chave permite que uma classe acesse a implementação da sua superclasse?
  - a) ☐ parent
  - b) ☒ super
  - c) ☐ base
  - d) ☐ this
  
3. Em Java, uma classe pode herdar de quantas superclasses diretamente?
  - a) ☒ Apenas 1
  - b) ☐ 2
  - c) ☐ 3
  - d) ☐ Quantas forem necessárias
  
4. O que significa encapsulamento em Java?
  - a) ☐ Reutilização de código por meio da herança
  - b) ☐ Agrupamento de classes em pacotes
  - c) ☒ Proteção dos atributos através de modificadores de acesso e métodos
  - d) ☐ Organização de código em interfaces

5. Qual modificador de acesso permite que um atributo seja acessado apenas dentro da própria classe?

- a) ☐ public
- b) ☐ protected
- c) ☒ private
- d) ☐ default

6. O modificador de acesso **protected** permite acesso:

- a) ☐ Apenas na mesma classe
- b) ☐ Apenas no mesmo pacote
- c) ☒ Na mesma classe, no mesmo pacote e por subclasses
- d) ☐ Em qualquer lugar do programa

7. Em herança, se uma subclasse precisa chamar o construtor da superclasse, qual palavra-chave deve ser usada?

- a) ☐ base()
- b) ☒ super()
- c) ☐ parent()
- d) ☐ extends()

8. O que acontece se uma classe filha define um método com a mesma assinatura de um método da classe pai?

- a) ☐ Sobrecarga (overloading)
- b) ☒ Sobrescrita (overriding)
- c) ☐ Encapsulamento
- d) ☐ Polimorfismo estático

9. Em relação à herança, qual é a diferença entre `extends` e `implements`?

- a) ☒ `extends` é usado para herdar classes, `implements` para interfaces

- b) ☐ ``extends`` é usado para interfaces, ``implements`` para classes
- c) ☐ Ambos fazem a mesma coisa
- d) ☐ ``extends`` só funciona com classes abstratas

10. Se um atributo é declarado como ``private``, como ele pode ser acessado por outras classes?

- a) ☐ Diretamente pelo nome
- b) ☒ Através de métodos públicos getters e setters
- c) ☐ Apenas com herança
- d) ☐ Não pode ser acessado em nenhuma hipótese

11. Uma classe declarada como ``final``:

- a) ☐ Não pode ser instanciada
- b) ☒ Não pode ser herdada
- c) ☐ Não pode ter atributos
- d) ☐ Não pode ter métodos

12. Em Java, qual anotação é usada para indicar que um método sobrescreve um da superclasse?

- a) ☒ `@Override`
- b) ☐ `@Overload`
- c) ☐ `@Overwritten`
- d) ☐ `@Inherited`

13. É possível herdar múltiplas classes diretamente em Java?

- a) ☐ Sim, sempre
- b) ☐ Sim, mas apenas classes abstratas
- c) ☒ Não, apenas uma classe pode ser herdada diretamente
- d) ☐ Sim, mas somente no mesmo pacote

14.O encapsulamento promove qual dos seguintes benefícios?

- a) ☐ Acesso irrestrito a todos os atributos
- b) ☒ Segurança e controle sobre os dados da classe
- c) ☐ Permite múltipla herança
- d) ☐ Reduz a necessidade de construtores

15. Qual dos seguintes é um exemplo correto de herança em Java?

- a) ☒ `class Car extends Vehicle { }`
- b) ☐ `class Car implements Vehicle { }`
- c) ☐ `class Car inherit Vehicle { }`
- d) ☐ `class Car super Vehicle { }`

## Questões sobre Java (AWT, Swing e JavaFX)

1. O que significa a sigla AWT em Java?

- a) ☒ Abstract Window Toolkit
- b) ☐ Advanced Window Tools
- c) ☐ Application Window Toolkit
- d) ☐ Active Window Technology

2. Qual classe do AWT representa uma janela principal?

- a) ☒ Frame
- b) ☐ Window
- c) ☐ Dialog
- d) ☐ Panel

3. No AWT, qual componente é utilizado para capturar entrada de texto de uma única linha?

- a) ☐ TextArea
- b) ☒ TextField
- c) ☐ Label
- d) ☐ Button

4. Em Swing, todos os componentes são derivados de qual classe base?

- a) ☐ Component
- b) ☒ Jcomponent
- c) ☐ Container
- d) ☐ JFrame

5. Qual é a diferença principal entre AWT e Swing?

- a) ☐ Swing é multiplataforma, AWT não é

- b) ☒ (X) Swing é baseado em AWT, mas é leve e mais personalizável
- c) ☐ ( ) AWT possui mais componentes que Swing
- d) ☐ ( ) AWT é mais rápido que Swing

6. Em Swing, qual componente representa uma caixa de seleção (checkbox)?

- a) ☒ (X) JCheckBox
- b) ☐ ( ) JRadioButton
- c) ☐ ( ) JComboBox
- d) ☐ ( ) JTextField

7. O método `setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)` em um `JFrame` define:

- a) ☐ ( ) O tamanho inicial da janela
- b) ☒ (X) A ação ao clicar no botão de fechar
- c) ☐ ( ) A cor de fundo da janela
- d) ☐ ( ) O título da janela

8. Em JavaFX, qual classe é obrigatória para iniciar uma aplicação gráfica?

- a) ☒ (X) Application
- b) ☐ ( ) Stage
- c) ☐ ( ) Scene
- d) ☐ ( ) Node

9. Em JavaFX, o objeto `Stage` representa:

- a) ☒ (X) Um container de elementos gráficos (janela principal)
- b) ☐ ( ) Uma cena dentro da janela
- c) ☐ ( ) Um botão interativo
- d) ☐ ( ) Um layout de grid

10. No JavaFX, qual é o papel da classe `Scene`?

- a) ☐ Controlar a execução da aplicação
- b) ☒ Definir a estrutura e os elementos exibidos no Stage
- c) ☐ Carregar bibliotecas externas
- d) ☐ Manipular eventos de mouse e teclado

11. Para criar uma aplicação JavaFX, qual método da classe `Application` deve ser sobrescrito?

- a) ☒ `start(Stage primaryStage)`
- b) ☐ `init()`
- c) ☐ `launch(String[] args)`
- d) ☐ `main(String[] args)`

12. No Swing, qual layout organiza componentes em linha, da esquerda para a direita?

- a) ☒ `FlowLayout`
- b) ☐ `BorderLayout`
- c) ☐ `GridLayout`
- d) ☐ `BoxLayout`

13. Em JavaFX, qual pacote contém os controles de interface (como `Button`, `Label`, `TextField`)?

- a) ☐ `javafx.stage`
- b) ☒ `javafx.scene.control`
- c) ☐ `javafx.scene.layout`
- d) ☐ `javafx.event`

14. Qual destes é um exemplo de evento em JavaFX?

- a) ☒ `ActionEvent`
- b) ☐ `WindowListener`

c) ( ) KeyAdapter

d) ( ) MouseMotionListener

15. Qual biblioteca Java foi criada mais recentemente para GUIs e substitui gradualmente Swing e AWT?

a) ( ) SWT

b) (X) JavaFX

c) ( ) JDK UI

d) ( ) JApplet