

1. Criando uma classe simples:
 - Crie uma classe Carro com os atributos marca, modelo e ano.
 - Crie um objeto no main() e exiba suas informações.
 2. Criando uma classe com método:
 - Modifique a classe Carro para incluir um método `exibirInformacoes()`.
 - Esse método deve imprimir os atributos do carro.
 3. Criando múltiplos objetos:
 - No main(), crie três objetos Carro com valores diferentes e exiba suas informações.
 4. Criando um construtor:
 - Adicione um construtor à classe Carro que recebe marca, modelo e ano.
 - Use-o para criar objetos no main().
-

Encapsulamento: Getters e Setters

5. Tornando atributos privados:
 - Modifique a classe Carro para que seus atributos sejam `private`.
6. Criando métodos get e set:
 - Adicione métodos para acessar e modificar os atributos privados.
7. Validação em setters:
 - No método `setAno(int ano)`, impeça valores menores que 1886.
8. Criando uma classe Pessoa com encapsulamento:
 - Atributos privados: nome, idade, altura.
 - Métodos get e set para cada atributo.
9. Validação no `setIdade()`:
 - Não permitir valores negativos para idade.
10. Criando uma classe `ContaBancaria`:

- Atributos privados: titular, saldo.
 - Métodos: depositar(), sacar(), getSaldo().
-

Modificadores de Acesso

11. Criando uma classe Produto:

- Atributos: nome, preco, quantidade.
- Modificadores de acesso: private.
- Métodos public para manipular o estoque.

12. Testando modificadores em outra classe:

- Tente acessar os atributos privados diretamente e veja o erro.

13. Criando um atributo protected:

- Adicione protected int codigo à classe Produto.
- Crie uma subclasse e tente acessá-lo.

14. Modificador default (sem modificador):

- Crie um atributo sem modificador e tente acessá-lo de outra classe no mesmo pacote.

15. Atributo static e acesso sem instância:

- Adicione um atributo static int totalProdutos à classe Produto.
 - Incremente esse valor sempre que um novo objeto for criado.
-

Herança e Encapsulamento

16. Criando uma classe Animal:

- Atributos private: nome, idade.
- Métodos get e set.
- Método emitirSom().

17. Criando subclasses Cachorro e Gato:

- Herde de Animal.
- Sobrescreva emitirSom() para exibir "Latindo" e "Miando".

18. Modificando atributos herdados:

- Na subclasse, tente acessar diretamente os atributos private da superclasse.

19. Usando super para chamar métodos da superclasse:

- No construtor das subclasses, use super(nome, idade).

20. Criando uma classe Funcionario:

- Atributos private: nome, salarioBase.
- Métodos: getNome(), getSalarioBase().
- Subclasses Gerente e Programador.

21. **Criando um sistema de cadastro:**

- Classe Cliente: atributos nome, email.
- Classe Pedido: Cliente cliente, double valorTotal.
- Relacione Cliente com Pedido e exiba os dados.

22. **Criando uma classe Retângulo:**

- Atributos privados: largura, altura.
- Métodos calcularArea() e calcularPerimetro().

23. **Criando uma classe Aluno:**

- Atributos private: nome, matricula, nota1, nota2.
- Método calcularMedia().

24. **Criando um sistema de conta bancária com validação:**

- Atributos saldo e titular.
- Impedir saque se não houver saldo suficiente.

25. **Criando um sistema de estoque:**

- Classe Produto: nome, quantidade.
- Classe Estoque: método verificarDisponibilidade(nome).