

Atividade Teórica

- 1. Qual é a principal diferença entre programação estruturada e programação orientada a objetos?**
 - a) A programação estruturada não utiliza funções, enquanto a programação orientada a objetos utiliza.
 - b) A programação estruturada se concentra em objetos, enquanto a POO se concentra em funções.
 - c) A programação estruturada se baseia em estruturas de controle, enquanto a POO se baseia em classes e objetos.
 - d) A programação estruturada é mais eficiente que a POO.
- 2. Quais são as três estruturas básicas da programação estruturada?**
 - a) Funções, classes e objetos
 - b) Sequência, decisão e repetição
 - c) Atributos, métodos e interfaces
 - d) Encapsulamento, herança e polimorfismo
- 3. Na Programação Orientada a Objetos (POO), o que é um objeto?**
 - a) Uma função que realiza uma tarefa específica
 - b) Uma instância de uma classe
 - c) Um conjunto de variáveis
 - d) Uma estrutura de repetição
- 4. Qual das opções a seguir define corretamente o conceito de encapsulamento na POO?**
 - a) Herança de métodos e atributos de outra classe
 - b) Habilidade de usar métodos com o mesmo nome
 - c) Ocultação de detalhes internos de uma classe e controle de acesso aos seus dados
 - d) Definição de métodos que devem ser implementados por outras classes
- 5. Qual é a função do polimorfismo na POO?**
 - a) Permitir que classes diferentes tenham métodos diferentes
 - b) Permitir que objetos de classes diferentes sejam tratados como objetos de uma classe comum
 - c) Proteger os atributos de uma classe
 - d) Facilitar a reutilização de código
- 6. O que são getters e setters na POO?**
 - a) Métodos que definem os comportamentos de uma classe
 - b) Métodos que acessam e modificam os atributos de uma classe

- c) Atributos de uma classe
- d) Classes que herdam de outras classes

7. Qual a principal vantagem da herança na POO?

- a) Evitar a necessidade de criar classes
- b) Permitir a criação de métodos duplicados
- c) Facilitar a reutilização de código ao permitir que classes compartilhem atributos e métodos
- d) Fornecer segurança aos dados de uma classe

8. Qual das seguintes afirmações sobre interfaces é verdadeira?

- a) As interfaces definem a implementação dos métodos que as classes devem ter.
- b) As interfaces descrevem apenas o que uma classe deve fazer, não como deve fazer.
- c) As interfaces servem para ocultar os dados de uma classe.
- d) As interfaces são usadas para criar objetos diretamente.

9. Como seria o uso da programação estruturada para calcular a área de um quadrado?

- a) Criar uma classe com um método que calcula a área.
- b) Criar uma função que recebe o lado do quadrado e retorna a área.
- c) Definir uma interface para calcular a área.
- d) Usar um loop para calcular a área.

10. Em qual cenário a POO seria mais vantajosa que a programação estruturada?

- a) Em programas simples, como uma calculadora básica.
- b) Em projetos grandes e complexos que exigem organização e reutilização de código.
- c) Em scripts que precisam de loops intensos.
- d) Quando o foco é apenas em cálculos matemáticos.



Atividade Prática

Parte 1: Programação Estruturada

1. Sequência, Decisão e Repetição:

- Crie um programa que:
 - Receba a idade de uma pessoa e, com base nisso, exiba se ela é menor de idade (menor que 18), maior de idade (entre 18 e 65) ou idosa (acima de 65).
 - Calcule a soma dos números de 1 a 100 usando um loop

Parte 2: Programação Orientada a Objetos

2. Criando Classes e Objetos:

- Crie uma classe chamada Carro que possui os seguintes atributos:
 - marca
 - modelo
 - ano
 - velocidade_atual
- A classe deve ter métodos para:
 - Acelerar (acelerar()) - aumenta a velocidade em 10.
 - Frear (frear()) - reduz a velocidade em 10.
 - Mostrar as informações do carro (mostrar_info())

Parte 3: Polimorfismo e Herança

3. Exemplo Prático com Polimorfismo e Herança:

- Crie uma classe Animal com um método emitir_som.
- Crie duas subclasses, Cachorro e Gato, que herdam de Animal e sobrescrevem o método emitir_som para latir e miar, respectivamente.

