Programação Orientada a Objetos

**Instruções:** Escolha a melhor resposta para cada questão.

# Tópicos:

* Pilares da POO
* Classes
* Objetos
* Propriedades

# Questões:

1. **Qual dos seguintes NÃO é um pilar da Programação Orientada a Objetos (POO)?**
   1. Abstração
   2. Encapsulamento
   3. Modularidade OK
   4. Herança
   5. Polimorfismo

# Qual conceito da POO permite que uma classe herde atributos e métodos de outra classe?

* 1. Abstração
  2. Encapsulamento
  3. Herança OK
  4. Polimorfismo
  5. Modularidade

# Qual conceito da POO permite que um método tenha diferentes implementações em classes diferentes, mas com a mesma assinatura?

* 1. Abstração
  2. Encapsulamento
  3. Herança
  4. Polimorfismo OK
  5. Modularidade

# Qual o principal objetivo da abstração na POO?

* 1. Ocultar detalhes de implementação OK
  2. Criar novas classes a partir de classes existentes
  3. Definir métodos com diferentes implementações
  4. Criar objetos a partir de classes
  5. Agrupar métodos e atributos em uma estrutura lógica

# O que é uma classe na POO?

* 1. Uma instância de um objeto OK
  2. Um conjunto de instruções para criar objetos
  3. Um conjunto de métodos
  4. Um conjunto de propriedades
  5. Um objeto específico

# O que é um objeto na POO?

* 1. Uma instância de uma classe
  2. Um conjunto de instruções para criar objetos
  3. Um conjunto de métodos
  4. Um conjunto de propriedades OK
  5. Uma classe específica

# Qual é a relação entre classes e objetos na POO?

* 1. Classes são instâncias de objetos OK
  2. Objetos são instâncias de classes
  3. Classes e objetos são sinônimosX’
  4. Não existe relação entre classes e objetos
  5. Objetos são a estrutura base para a criação de classes

# Qual o principal objetivo do encapsulamento na POO?

* 1. Ocultar detalhes de implementação de uma classe OK
  2. Criar novas classes a partir de classes existentes
  3. Definir métodos com diferentes implementações
  4. Criar objetos a partir de classes
  5. Agrupar métodos e atributos em uma estrutura lógica

# O que são propriedades em uma classe na POO?

* 1. Métodos que definem o comportamento de um objeto
  2. Variáveis que armazenam dados de um objeto OK
  3. Classes que definem a estrutura de um objeto
  4. Objetos que representam a instância de uma classe
  5. Métodos que interagem com outras classes

# Qual é a diferença entre métodos e propriedades em uma classe?

* 1. Métodos definem o comportamento, enquanto propriedades armazenam dados OK
  2. Propriedades definem o comportamento, enquanto métodos armazenam dados -
  3. Métodos e propriedades são sinônimos
  4. Não existe diferença entre métodos e propriedades
  5. Métodos são usados para criar objetos, enquanto propriedades são usadas para interagir com objetos

# Qual o conceito da POO que permite a reutilização de código?

* 1. Abstração
  2. Encapsulamento
  3. Herança
  4. Polimorfismo OK
  5. Modularidade

# Qual dos seguintes é um exemplo de polimorfismo na POO?

* 1. Criar uma classe com o mesmo nome de outra classe
  2. Criar um método com o mesmo nome em classes diferentes OK
  3. Criar uma classe que herda atributos e métodos de outra classe
  4. Criar uma classe com métodos abstratos
  5. Criar uma classe com um atributo privado

# Qual o principal benefício da POO?

* 1. Facilita a escrita de programas complexos
  2. Permite a criação de software mais eficiente
  3. Promove o reuso de código
  4. Aumenta a segurança do software
  5. Todas as alternativas acima OK

# Qual o papel das interfaces na POO?

* 1. Definir o comportamento de uma classe
  2. Criar objetos a partir de classes
  3. Implementar métodos abstratos
  4. Definir contratos que as classes devem seguir OK
  5. Implementar o encapsulamento

# Qual dos seguintes é um exemplo de abstração na POO?

* 1. Criar uma classe com o mesmo nome de outra classe
  2. Criar um método com o mesmo nome em classes diferentes
  3. Criar uma classe que herda atributos e métodos de outra classe
  4. Criar uma classe com métodos abstratos OK
  5. Criar uma classe com um atributo privado

# Qual a diferença entre uma classe abstrata e uma interface?

* 1. Classes abstratas podem ter implementações, enquanto interfaces não OK
  2. Interfaces podem ter implementações, enquanto classes abstratas não
  3. Classes abstratas e interfaces são sinônimos
  4. Não existe diferença entre classes abstratas e interfaces
  5. Classes abstratas são usadas para criar objetos, enquanto interfaces são usadas para definir contratos

# Qual o papel dos construtores na POO?

* 1. Inicializar os atributos de um objeto OK
  2. Criar um objeto a partir de uma classe
  3. Definir o comportamento de um objeto
  4. Implementar métodos abstratos
  5. Implementar o encapsulamento

# Qual a diferença entre atributos e métodos em uma classe?

* 1. Atributos definem o comportamento de um objeto, enquanto métodos armazenam dados
  2. Métodos definem o comportamento de um objeto, enquanto atributos armazenam dados OK
  3. Atributos e métodos são sinônimos
  4. Não existe diferença entre atributos e métodos
  5. Atributos são usados para criar objetos, enquanto métodos são usados para interagir com objetos

# Qual o papel do encapsulamento na segurança do software?

* 1. O encapsulamento impede o acesso direto aos dados de um objeto OK
  2. O encapsulamento permite o acesso direto aos dados de um objeto
  3. O encapsulamento não tem relação com a segurança do software
  4. O encapsulamento aumenta o risco de erros no software
  5. O encapsulamento diminui a performance do software

# Qual é o principal objetivo da modularidade na POO?

* 1. Dividir o código em partes menores e mais fáceis de gerenciar OK
  2. Criar novas classes a partir de classes existentes
  3. Definir métodos com diferentes implementações
  4. Criar objetos a partir de classes
  5. Agrupar métodos e atributos em uma estrutura lógica