

**As questões abaixo são um teste, deve-se responder o gabarito ao fim da folha e devolver para o e-mail [dannel\\_kayke@hotmail.com](mailto:dannel_kayke@hotmail.com) até as 12h do dia 18/11/2023.**

**1. Qual é a principal vantagem da orientação a objetos?**

- a) Menor reutilização de código
- b) Melhor desempenho
- c) Encapsulamento e reutilização de código
- d) Acesso direto aos dados
- e) Uso limitado de classes

**2. Em TypeScript/JavaScript, qual palavra-chave é usada para criar uma classe?**

- a) type
- b) interface
- c) class
- d) struct
- e) classof

**3. O que é encapsulamento em programação orientada a objetos?**

- a) Acesso direto a todos os atributos da classe
- b) Ocultar detalhes internos e expor apenas o necessário
- c) Utilização exclusiva de interfaces
- d) Ausência de herança
- e) Uso de objetos literais

**4. Como se define um método em uma classe TypeScript/JavaScript?**

- a) function
- b) def
- c) método
- d) func
- e) method

**5. O que é herança?**

- a) Uso exclusivo de interfaces
- b) Capacidade de uma classe herdar propriedades e métodos de outra
- c) Restrição total ao acesso de atributos
- d) Uso de tipos genéricos
- e) Ausência de interfaces

**6. Qual é o operador utilizado para acessar um membro de uma classe em TypeScript?**

- a) .
- b) ->
- c) ::
- d) ::
- e) =>

**7. O que são propriedades estáticas em uma classe?**

- a) Propriedades que não podem ser alteradas
- b) Propriedades que pertencem à instância da classe
- c) Propriedades que pertencem à classe, não à instância
- d) Propriedades que são acessíveis apenas dentro da classe
- e) Propriedades que não podem ser acessadas diretamente

**8. Em TypeScript, como você evita que uma classe seja instanciada diretamente?**

- a) Utilizando a palavra-chave const
- b) Utilizando a palavra-chave abstract
- c) Marcando todos os métodos como privados
- d) Não é possível evitar a instanciação direta
- e) Utilizando a palavra-chave sealed

**9. O que é polimorfismo?**

- a) Capacidade de uma classe ter múltiplos construtores
- b) Capacidade de uma classe herdar de várias classes
- c) Capacidade de uma classe implementar várias interfaces
- d) Capacidade de uma classe ter métodos com o mesmo nome, mas com diferentes implementações
- e) Restrição ao uso de tipos genéricos

**10. Em TypeScript, como você declara uma interface?**

- a) declare interface Nome
- b) create interface Nome
- c) interface Nome
- d) new interface Nome
- e) type interface Nome

**11. O que são generics em TypeScript?**

- a) Tipos específicos que podem ser usados em qualquer contexto
- b) Tipos que podem ser usados apenas em funções
- c) Tipos que podem ser usados apenas em classes
- d) Tipos que podem ser usados apenas em interfaces
- e) Tipos que podem ser parametrizados para fornecer flexibilidade em tipos de dados

**12. Como você implementa a herança em TypeScript usando classes?**

- a) extends
- b) inherits
- c) implement
- d) inherit
- e) derive

**13. Qual é o propósito do construtor em uma classe?**

- a) Definir as propriedades estáticas da classe
- b) Inicializar instâncias da classe com valores iniciais
- c) Declarar métodos privados da classe
- d) Impor restrições de acesso aos métodos
- e) Criar uma nova instância da classe

**14. O que é um método abstrato em TypeScript?**

- a) Um método que não tem implementação e deve ser implementado nas classes derivadas
- b) Um método que não pode ser acessado externamente
- c) Um método que não pode ser sobrescrito
- d) Um método que só pode ser chamado internamente
- e) Um método que não pode ser declarado em uma interface

**15. Qual é a diferença entre uma classe abstrata e uma interface em TypeScript?**

- a) Uma classe abstrata pode ter implementações de método, enquanto uma interface não pode
- b) Uma interface pode ter propriedades, enquanto uma classe abstrata não pode
- c) Ambas podem ter implementações de método, mas apenas uma interface pode ter propriedades
- d) Ambas são usadas apenas para definir contratos e não podem ter implementações
- e) Uma classe abstrata pode ser instanciada diretamente, enquanto uma interface não pode

**16. O que é sobrecarga de método?**

- a) Ter vários métodos com o mesmo nome, mas com diferentes tipos de parâmetros
- b) Ter vários métodos com a mesma implementação
- c) Ter apenas um método em uma classe
- d) Ter métodos apenas em interfaces
- e) Ter métodos apenas em classes abstratas

**17. Como você instancia uma classe em TypeScript/JavaScript?**

- a) `new MyClass();`
- b) `MyClass.instance();`
- c) `create MyClass();`
- d) `instanceOf MyClass;`
- e) `MyClass.create();`

**18. O que é uma propriedade readonly em TypeScript?**

- a) Uma propriedade que não pode ser lida
- b) Uma propriedade que não pode ser modificada após a inicialização
- c) Uma propriedade que não pode ser declarada em interfaces
- d) Uma propriedade que só pode ser acessada por métodos internos da classe
- e) Uma propriedade que só pode ser acessada por outras classes da mesma hierarquia

	A	B	C	D	E
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					