WAYGROUND formerly Quizizz **Folhas de exercícios**

Nome

Turma

Data

2025.02 - MDCOO - Revisão - 2ºB

Total questões: 35

Tempo da folha de exercícios: 53minutos

Nome do instrutor: Fernando Alves

1. Dentro da temática de Orientação a Objetos, pode-se definir o seguinte exemplo: dada a classe CarroDePasseio, ela representa todos os atributos e métodos dos carros de passeio. Assim, essa classe consiste em um modelo para reservar um espaço de memória para manter e transformar os dados. A partir desse exemplo, assinale a alternativa que apresenta corretamente os conceitos de Atributo, Método e Instanciação.

   a) Atributos são o comportamento que os elementos podem adotar, Método é o conjunto de características de um dado Elemento e Instanciação é a capacidade de criar classes a partir de um objeto.

   b) Atributos são o conjunto de exceções para criação de um Elemento, Método é o passo a passo para construir a classe e Instanciação é a capacidade de conexão entre outras classes.

   c) Atributos são a capacidade que um Elemento tem de criar informações diferentes da Classe pai, Método é a característica que a classe tem de se comunicar com elementos externos e Instanciação é a capacidade da classe corrigir seus próprios erros.

   d) Atributos são o conjunto de características de um dado Elemento, Método é o comportamento que os elementos podem adotar e Instanciação é a capacidade de criar objetos a partir de uma classe.

2. Na orientação a objetos, há o conceito de encapsulamento que estabelece que

   a) classes com métodos encapsulados não admitem ter subclasses.

   b) aspectos internos de implementação de uma classe não são visíveis por outras classes.

   c) subclasses assumem características das superclasses correspondentes.

   d) classes com métodos encapsulados não podem possuir atributos.

3. João, Maria e José são clientes de um banco. O sistema de informatização do banco é capaz de cadastrar tais pessoas empregando características particulares como nome, endereço e outras informações julgadas relevantes. Além disso, o sistema pode realizar o cadastro ou a exclusão do cliente.

   Dentro do conceito de programação orientada a objetos

   I. João, Maria e José são exemplos de classes de um objeto que pode ser denominado CLIENTE;
   II. as características que definem João, Maria e José são denominados atributos;
   III. as operações de cadastro e exclusão de clientes são métodos implementados na classe.

   a) se somente as afirmativas I e II estiverem corretas
   b) se somente as afirmativas II e III estiverem corretas
   c) se somente a afirmativa I estiver correta
   d) se somente a afirmativa II estiver correta

4. No domínio da orientação a objetos, a ideia de existência de dados e funcionalidades acessados somente pelos objetos de uma classe, de forma interna, e de que deva existir alguma forma protegida de acesso a esses dados e funcionalidades, de modo que objetos de classes externas os enxerguem, está associada aos conceitos, respectivamente, de:

   a) encapsulamento e interface.
   b) polimorfismo e virtualidade.
   c) atributo e método.
   d) associação e herança.

5. Considere o cenário onde uma Classe B lega suas estruturas e comportamentos de uma Classe A. Essa relação entre a Classe A e a Classe B é caracterizada por:

   a) uma herança
   b) uma instância da classe
   c) um polimorfismo
   d) uma instância do objeto

6. Um dos conceitos do paradigma orientado a objetos consiste na alteração do funcionamento interno de um método herdado de um objeto pai. Assinale a alternativa correta que apresenta este conceito.

   a) Encapsulamento
   b) Herança
   c) Abstração
   d) Polimorfismo

7. Considere as seguintes afirmações sobre alguns fundamentos de Programação Orientada a Objetos.
I - Classe é um conceito orientado a objeto que encapsula dados e abstrações procedurais necessárias para descrever o conteúdo e o comportamento de alguma entidade do mundo real. Pode-se dizer que classe é uma descrição generalizada que descreve uma coleção de objetos similares.

II - Superclasse é a generalização de um conjunto de classes a ela relacionadas.
III - Subclasse é uma especialização da superclasse. Uma subclasse herda todos os atributos e operações associadas à sua superclasse e não pode incorporar atributos ou operações adicionais específicos.

a) Apenas I.
b) Apenas I e III.
c) I, II e III
d) Apenas I e II.

8. Em um programa, desenvolvido com uma linguagem orientada a objetos, uma classe Turma possui, como atributos, um professor, objeto da classe Professor e uma coleção de alunos, que são objetos da classe Aluno. Objetos das classes Aluno e Professor existem independente da existência de um objeto da classe Turma. A associação entre uma turma e objetos das classes Professor e Aluno é definido como

a) polimorfismo.
b) composição.
c) herança.
d) agregação.

9. Qual a propriedade, típica da orientação a objeto, que habilita uma quantidade de operações diferentes a ter o mesmo nome, diminuindo o acoplamento entre objetos?

a) Especialização
b) Polimorfismo
c) Encapsulamento
d) Herança

10. Na orientação a objetos, o conceito de encapsulamento corresponde à propriedade de

a) usar variáveis e constantes do tipo inteiro nos métodos das classes implementadas.
b) ter um conjunto de objetos com a mesma classe.
c) receber, por uma classe, uma mensagem sem parâmetros
d) esconder ou ocultar detalhes da implementação de uma dada classe de outras classes.

11. Na programação orientada a objetos, a instanciação de objetos tem o objetivo de:

a) inicializar uma classe
b) definir atributos e métodos de uma classe
c) abstrair os atributos de um objeto
d) criar um objeto

12. Associe os Princípios de Orientação a Objetos com os seus significados:

Conjunto de objetos que possuem o mesmo tipo.	◌	◌ Abstração
Ocultamento de partes da classe que não podem ser acessada diretamente por outras.	◌	◌ Classe
Define que uma determinada operação pode se comportar de diferentes formas em diferentes classes.	◌	◌ Polimorfismo
Elucida apenas as propriedades comuns de um conjunto de objetos, omitindo os detalhes.	◌	◌ Herança
Permite ao usuário definir tipos de forma incremental, a partir de tipos existentes.	◌	◌ Encapsulamento

13. A forma de reutilização de *software* em que novas classes adquirem os membros de outras já existentes e aprimoram essas classes com novas capacidades é denominada

a) herança.
b) polimorfismo.
c) tratamento de exceções.
d) encapsulamento.

14. Procedures, funções e subrotinas são conceitos das técnicas de programação tradicionais que correspondem, nas técnicas orientadas ao objeto:

a) aos métodos.
b) às variáveis de instância.
c) à classe.
d) à hereditariedade.

15. Na programação orientada a objetos, os métodos representam.

a) o tipo de herança existente entre as classes.
b) os tipos de linguagens de programação utilizados
c) a implementação das ações das classes definidas.
d) as associações estabelecidas entre as classes

16. Na Programação orientada a objetos, é o princípio que oferece a capacidade de um método pode ser implementado de diferentes formas, ou mesmo de realizar coisas diferentes, ou seja, um único serviço pode oferecer variações, conforme se aplique a diferentes subclasses de uma superclasse.

a) Abstração.
b) Encapsulamento.
c) Polimorfismo.
d) Herança.

17. No âmbito dos princípios de concepção e programação orientada a objeto, é correto afirmar que "um objeto da subclasse é um objeto da superclasse, ou seja, os objetos da subclasse podem ser tratados como objetos da superclasse". Esta afirmação é possível quando se refere ao contexto de

a) Encapsulamento.
b) Herança.
c) Polimorfismo.
d) Abstração.

18. Um subconjunto de entidades, dentro de um conjunto de entidades, que tem atributos distintos das demais entidades do conjunto (refinamento em subgrupos topdown) denomina-se

a) generalização.
b) herança.
c) sistematização.
d) especialização.

19. Uma técnica que consiste em separar aspectos externos dos internos da implementação de um objeto, isto é, determinados detalhes ficam ocultos aos demais objetos e dizem respeito apenas ao próprio objeto. Trata-se de:

a) polimorfismo.
b) herança.
c) generalização.
d) encapsulamento.

20. Na programação orientada a objetos, há os métodos chamados de construtores, sobre os quais é correto afirmar que

a) sempre retornam um dos dois seguintes valores: falso ou verdadeiro.
b) há, obrigatoriamente, pelo menos dois construtores em cada classe
c) são executados sempre que um atributo de um dado objeto for consultado.
d) são métodos executados na criação do objeto

21. São considerados princípios do Paradigma da Orientação a Objetos:

a) Abstração (Classes, Objetos, Atributos e Métodos), Encapsulamento, Herança e Polimorfismo.

b) Dados, Funções, Encapsulamento e Hereditariedade.

c) Unicamente Classes, Encapsulamento, Hereditariedade e polimorfismo.

d) Abstrações (Classes, Dados, Atributos dos dados e Funções), Encapsulamento e Diagramas de Classe.

22. Considerando o paradigma da orientação a objeto, para o enunciado abaixo, assinale a alternativa correta.
Paulo pode andar, correr, pular. Ele tem 30 anos, é casado e trabalha com Tecnologia da Informação e Comunicação.

a) Ter 30 anos, ser casado, andar e correr são estados que podem ser associados ao objeto Paulo.

b) Paulo é um objeto. Tem como métodos andar, correr e pular e, 30 anos, ser casado e trabalhar com TICs são atributos.

c) Paulo é uma classe pois há outros Paulos que tem idade diferente de 30 anos e que exercem outras profissões.

d) Andar, correr e pular, são objetos que podem ter como variável Paulo.

23. Marque a alternativa que NÃO representa uma característica da orientação a objetos.

a) Encapsulamento.

b) Recursão.

c) Polimorfismo.

d) Reutilização de Código.

24. Na programação orientada a objeto,

a) um construtor serve para inicializar os atributos e é executado automaticamente sempre que ocorre a criação de um novo objeto.

b) encapsulamento consiste na aglutinação de aspectos internos e externos de um objeto.

c) atributos são classes que se encontram dentro de cada um dos objetos restritos a determinados tipos.

d) métodos são variáveis que se encontram dentro de cada um dos objetos de uma classe.

25. Em conformidade com a metodologia orientada a objetos, com a finalidade de evitar que partes de um programa se tornem tão independentes que uma pequena alteração tenha grandes efeitos em cascata, é aplicado um recurso que separa os aspectos externos e acessíveis de um objeto dos detalhes internos de implementação.

Esse recurso utiliza um princípio da *Orientação a Objetos* que propõe ocultar determinados elementos de uma classe das demais classes. O objetivo ao colocar uma proteção ao redor é prevenir contra os efeitos colaterais indesejados ao ter essas propriedades modificadas de forma inesperada.

Este recurso é conhecido por:

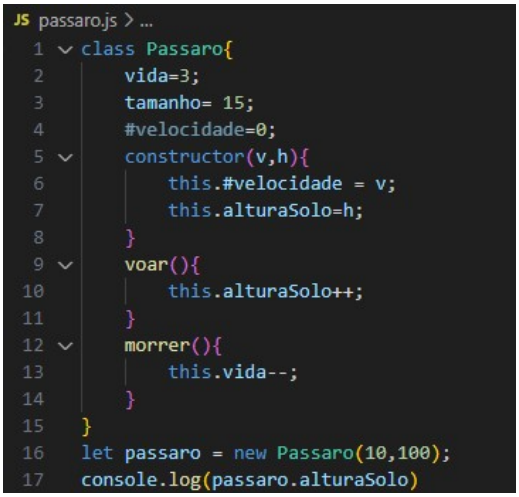
a) coesão.

b) polimorfismo.

c) acoplamento.

d) encapsulamento.

26.



```
JS passaro.js > ...
1  class Passaro{
2      vida=3;
3      tamanho= 15;
4      #velocidade=0;
5      constructor(v,h){
6          this.#velocidade = v;
7          this.alturaSolo=h;
8      }
9      voar(){
10         this.alturaSolo++;
11     }
12     morrer(){
13         this.vida--;
14     }
15 }
16 let passaro = new Passaro(10,100);
17 console.log(passaro.alturaSolo)
```

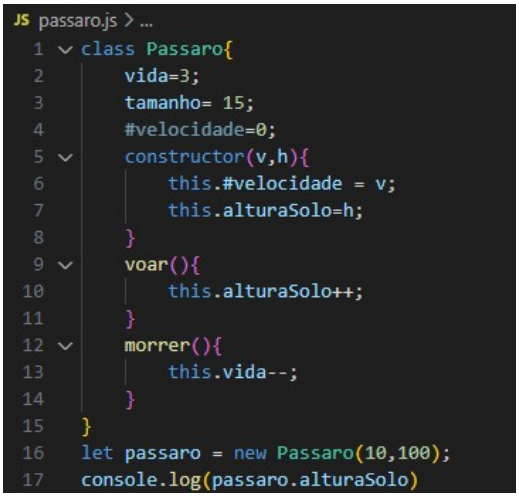
Organize essas opções nas categorias certas

Classifique o seguinte

Passaro | Passaro | Vida | Tamanho | Velocidade | AlturaSolo | Voar() | Morrer()

Objeto	Classe	Atributos	Métodos

27.



```
JS passaro.js > ...
1  class Passaro{
2      vida=3;
3      tamanho= 15;
4      #velocidade=0;
5      constructor(v,h){
6          this.#velocidade = v;
7          this.alturaSolo=h;
8      }
9      voar(){
10         this.alturaSolo++;
11     }
12     morrer(){
13         this.vida--;
14     }
15 }
16 let passaro = new Passaro(10,100);
17 console.log(passaro.alturaSolo)
```

Qual valor será impresso no console do navegador ao executar o trecho do código ao lado?

a) 100

b) 10

c) 3

d) 0

28.

```
JS passaro.js > ...
1  class Passaro{
2      vida=3;
3      tamanho= 15;
4      #velocidade=0;
5      constructor(v,h){
6          this.#velocidade = v;
7          this.alturaSolo=h;
8      }
9      voar(){
10         this.alturaSolo++;
11     }
12     morrer(){
13         this.vida--;
14     }
15 }
16 let passaro = new Passaro(10,100);
17 console.log(passaro.alturaSolo)
```

Qual atributo é privado?

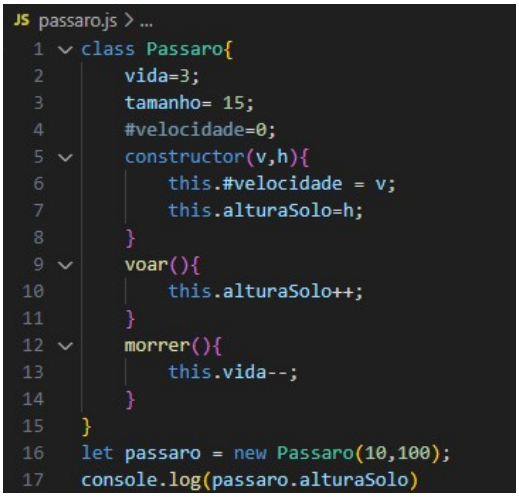
a) tamanho

b) vida

c) alturaSolo

d) velocidade

29.



```
JS passaro.js > ...
1  class Passaro{
2      vida=3;
3      tamanho= 15;
4      #velocidade=0;
5      constructor(v,h){
6          this.#velocidade = v;
7          this.alturaSolo=h;
8      }
9      voar(){
10         this.alturaSolo++;
11     }
12     morrer(){
13         this.vida--;
14     }
15 }
16 let passaro = new Passaro(10,100);
17 console.log(passaro.alturaSolo)
```

Como eu executo o método morrer ?

a) exec morrer

b) passaro.morrer

c) passaro.morrer()

d) call method morrer;

30.

```
JS passaro.js > ...
1  class Passaro{
2      vida=3;
3      tamanho= 15;
4      #velocidade=0;
5      constructor(v,h){
6          this.#velocidade = v;
7          this.alturaSolo=h;
8      }
9      voar(){
10         this.alturaSolo++;
11     }
12     morrer(){
13         this.vida--;
14     }
15 }
16 let passaro = new Passaro(10,100);
17 console.log(passaro.alturaSolo)
```

Considerando apenas o trecho do código ao lado, é possivel deduzir que:

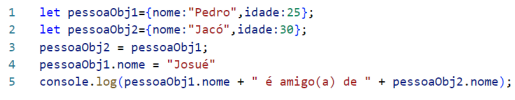
a) O objeto passaro voa a uma altura constante.

b) O objeto passaro está sempre parado.

c) O objeto passaro voa para baixo.

d) O objeto passaro voa para cima.

31.



```
1  let pessoaObj1={nome:"Pedro",idade:25};
2  let pessoaObj2={nome:"Jacó",idade:30};
3  pessoaObj2 = pessoaObj1;
4  pessoaObj1.nome = "Josué"
5  console.log(pessoaObj1.nome + " é amigo(a) de " + pessoaObj2.nome);
```

O resultado esperado é:

a) Jacó é amigo(a) de Pedro

b) Josué é amigo(a) de Pedro

c) Pedro é amigo(a) de Jacó

d) Josué é amigo(a) de Josué

32.

```
const pessoa={
    nome: "Pedro",
    idade: "18",
    timeFutebol:"Vasco",
    idolo:"Pedrinho",
    paixao:"Maria"
}
let paixao="timeFutebol"
let nome="Maria"
console.log(pessoa.nome+" é a apaixonado(a) pelo(a) " + pessoa[paixao])
console.log(pessoa['paixao']+" é a apaixonado(a) pelo(a) " + pessoa['nome'])
```

O resultado esperado é:

a) Pedro é a apaixonado(a) pelo(a) Vasco
Maria é a apaixonado(a) pelo(a) Pedro

b) Maria é a apaixonado(a) pelo(a) Pedrinho
Maria é a apaixonado(a) pelo(a) Pedro

c) Pedro é a apaixonado(a) pelo(a) Maria
Maria é a apaixonado(a) pelo(a) Pedro

d) Pedro é a apaixonado(a) pelo(a) Pedrinho
Vasco é a apaixonado(a) pelo(a) Pedro

33.

```
var pessoa2 ={
    nome: "Joana",
    idade: "18",
    timeFutebol:"Sem Time",
    idolo:"Anitta",
    paixao:"Pedro"
}
pessoa2.idolo="Beyoncé"
pessoa2.nome="Maria"
let paixao="idolo"
console.log(pessoa2.nome+" é a apaixonado(a) pelo(a) " + pessoa2['idolo'])
console.log(pessoa2[paixao]+" é a apaixonado(a) pelo(a) " + pessoa2['paixao'])
```

O resultado esperado é:

a) Joana é a apaixonado(a) pelo(a) Beyoncé
Maria é a apaixonado(a) pelo(a) Anitta

b) Joana é a apaixonado(a) pelo(a) Pedro
Anitta é a apaixonado(a) pelo(a) Pedro

c) Maria é a apaixonado(a) pelo(a) Pedro
Joana é a apaixonado(a) pelo(a) Pedro

d) Maria é a apaixonado(a) pelo(a) Beyoncé
Beyoncé é a apaixonado(a) pelo(a) Pedro

34.

```
class Forma{
    #lados;
    constructor(vetorLados){
        this.#lados = vetorLados;
    }
    calcular2p(){
        let perimetro=0
        for(let i=0;i<this.#lados.length;i++)
            perimetro+=this.#lados[i];
        return perimetro;
    }
    calcularArea(){
        return this.#lados[0]*this.#lados[0]
    }

    getLados(){
        return this.#lados
    }
}
const tri = new Forma([3,4,5]);
const qua = new Forma([4,4,4,4]);
console.log(tri.calcularArea());
console.log(qua.calcular2p());
```

O resultado esperado é:

a) 10
16

b) 9
16

c) 3
4

d) 6
12

35.

```
class Forma{
    constructor(vetorLados){
        this.lados = vetorLados;
    }
    calcular2p(){
        let perimetro=0
        for(let i=0;i<this.lados.length;i++)
            perimetro+=this.lados[i];
        return perimetro;
    }
    calcularArea(){
        return this.lados[0]*this.lados[0]
    }

}
const tri = new Forma([3,4,5]);
const qua = new Forma([4,4,4,4]);
qua.lados=[5,5,5,5]
console.log(tri.calcularArea());
console.log(qua.calcular2p());
```

O resultado esperado é:

a) 12
16

b) 6
5

c) 9
20

d) 12
20

Gabaritos

1. d) Atributos são o conjunto de características de um dado Elemento, Método é o comportamento que os elementos podem adotar e Instanciação é a capacidade de criar objetos a partir de uma classe.
2. b) aspectos internos de implementação de uma classe não são visíveis por outras classes.
3. b) se somente as afirmativas II e III estiverem corretas
4. a) encapsulamento e interface.
5. a) uma herança
6. d) Polimorfismo
7. d) Apenas I e II.
8. d) agregação.
9. b) Polimorfismo
10. d) esconder ou ocultar detalhes da implementação de uma dada classe de outras classes.
11. d) criar um objeto
12. Elucida apenas as propriedades comuns de um conjunto de objetos, omitindo os detalhes.
    - Abstração,
    Conjunto de objetos que possuem o mesmo tipo.
    - Classe,
    Define que uma determinada operação pode se comportar de diferentes formas em diferentes classes.
    - Polimorfismo,
    Permite ao usuário definir tipos de forma incremental, a partir de tipos existentes.
    - Herança,
    Ocultamento de partes da classe que não podem ser acessada diretamente por outras.
    - Encapsulamento

13. a) herança.
14. a) aos métodos.
15. c) a implementação das ações das classes definidas.
16. c) Polimorfismo.
17. b) Herança.
18. d) especialização.
19. d) encapsulamento.
20. d) são métodos executados na criação do objeto
21. a) Abstração (Classes, Objetos, Atributos e Métodos), Encapsulamento, Herança e Polimorfismo.
22. b) Paulo é um objeto. Tem como métodos andar, correr e pular e, 30 anos, ser casado e trabalhar com TICs são atributos.
23. b) Recursão.
24. a) um construtor serve para inicializar os atributos e é executado automaticamente sempre que ocorre a criação de um novo objeto.
25. d) encapsulamento.
26. **Objeto**

    Passaro

    Classe

    Passaro

    Atributos

    Vida | Tamanho

    Velocidade

    AlturaSolo

    Métodos

    Voar() | Morrer()
27. a) 100
28. d) velocidade
29. c) passaro.morrer()
30. d) O objeto passaro voa para cima.
31. d) Josué é amigo(a) de Josué
32. a) Pedro é a apaixonado(a) pelo(a) Vasco Maria é a apaixonado(a) pelo(a) Pedro
33. d) Maria é a apaixonado(a) pelo(a) Beyoncé Beyoncé é a apaixonado(a) pelo(a) Pedro
34. b) 9 16
35. c) 9 20