Resposta *poo\_01\_exercicio*

Germano Barbosa da Silva Júnior — 2023111TADS0015

**1. Qual a diferença entre objetos e classes? Exemplifique.**

Uma classe é a declaração, o tipo de objeto. Um objeto é uma instância específica de uma classe.

Exemplo: Cachorro, *Canis lupus familiaris*, é uma classe, Kelvin o cachorro é um objeto.

**2. De forma breve, conceitue atributos e métodos. Pesquise e exemplifique um exemplo de objeto que possua atributos e métodos (notação livre).**

Um atributo é um valor que pertence ao objeto, um método é uma função que tem uma referência implícita ao objeto, isso permite que o metodo leia e modifique os atributos da classe.

Exemplo: Eu tenho uma classe Rect, que tem os atributos posição, do tipo Point, e tamanho do tipo Size, um exemplo de método para essa classe seria o método contains que recebe como parâmetro uma instãncia de Point, e retorna True, caso o ponto esteja contido dentro do retângulo, e retorna False senão.

**3. A abstração visa focar no que é importante para um sistema. Você concorda que um atributo de uma pessoa pode ser importante ou não dependendo do contexto do sistema. Enumere na tabela abaixo contextos/sistemas distintos em que os atributos abaixo seriam relevantes:**

Quais atributos são necessários na implementação depende das funcionalidades que o sistema precisa ter.

|  |  |
| --- | --- |
| Atributo | Sistema em que é importante |
| Peso | Sistema hospitalar |
| Tipo de CNH | Sistema de aluguel de carros |
| Tipo Sanguíneo | Sistema hospitalar |
| Habilidade destra | Sistema escolar |
| Percentual de gordura | Sistema hospitalar |
| Saldo em conta | Sistema bancário |
| Etinia | Sistema de concurso |

**4. Considerando os objetos Pessoa e Conta:**

**a. Seria interessante em um sistema bancário um objeto "conta" possuir uma**

**"pessoa" como um atributo interno representando o titular da conta?**

Não, pois uma pessoa pode ter múltiplas contas, e dependendo do sistema, uma conta pode ser gerenciada por múltiplas pessoas. De qualquer forma, relação não é um para um.

**b. Olhando no sentido inverso, seria interessante uma pessoa possuir mais de**

**uma conta como atributo? Que elemento da programação estruturada melhor**

**representaria o conjunto de contas de uma pessoa?**

Dependendo das funcionalidades que os sistema precisa ter, sim.

Atributos, as várias contas da pessoa pode ser representada por um atributo contendo um array de contas, ou um array de referências as contas.

**5. Identifique pelo menos 5 objetos de um sistema de controle acadêmico. Ex: aluno.**

Aluno, Avaliação, Material, Curso, Nota, Atividade

**6. Imagine um jogo qualquer. Identifique o máximo de objetos possíveis e eventuais**

**características (atributos) e comportamentos (métodos) que os mesmos poderiam**

**ter.**

classe Jogador

atributo vida, número

atributo vida máxima, número

atributo munição, número

atributo munição máxima, número

atributo nome, texto

método receberDano

método completarVida

método completarMunicao

método estaVivo, retorna booleano

classe Inimigo

atributo vida, número

atributo vida máxima, número

atributo dano, número

atributo velocidade, número

atributo tipo, string

método receberDano

método executarAI

método estaVivo, retorna booleano

**Usando playground (https://www.typescriptlang.org/play), faça:**

**7. Considerando o exemplo da classe Retangulo dos slides, implemente um método adicional chamado que calcule o perímetro do retângulo e altere a classe TestaRetangulo para exibir o cálculo do perímetro.**

class Retangulo {

l1: number = 0;

l2: number = 0;

calcularArea(): number {

return this.l1 \* this.l2;

}

calcularPerimetro(): number {

return (this.l1 + this.l2) \* 2;

}

}

class TestaRetangulo {

testar() {

let retangulo: Retangulo;

retangulo = new Retangulo();

retangulo.l1 = 10;

retangulo.l2 = 20;

console.log("retangulo:", retangulo);

console.log("calcularArea:", retangulo.calcularArea());

console.log("calcularPerimetro:", retangulo.calcularPerimetro());

}

}

let testador: TestaRetangulo = new TestaRetangulo();

testador.testar();

**8. Crie uma classe Circulo que possua um atributo raio. Crie dois métodos que**

**calculam a área e o perímetro. Instancie um objeto dessa classe, atribua um valor**

**ao raio e exiba a área e o perímetro chamando os dois métodos definidos.**

class Circulo {

raio: number = 0;

calcularArea(): number {

return this.raio \* this.raio \* Math.PI ;

}

calcularPerimetro(): number {

return this.raio \* Math.PI \* 2;

}

}

let circulo = new Circulo;

circulo.raio = 10;

console.log("raio:", circulo.raio);

console.log("area:", circulo.calcularArea());

console.log("perímetro:", circulo.calcularPerimetro());

**9. Crie uma classe chamada SituacaoFinanceira com os atributos valorCreditos e**

**valorDebitos. Crie um método chamado saldo() que retorna/calcula a diferença**

**entre crédito e débito. Instancie uma classe SituacaoFinanceira, inicialize os dois**

**atributos e exiba o resultado do método saldo().**

class SituacaoFinanceira {

valorCreditos: number = 0;

valorDebitos: number = 0;

saldo(): number {

return this.valorCreditos - this.valorDebitos;

}

}

let situacaoFinanceira: SituacaoFinanceira = new SituacaoFinanceira;

situacaoFinanceira.valorCreditos = 100;

situacaoFinanceira.valorDebitos = 30;

console.log("creditos:", situacaoFinanceira.valorCreditos);

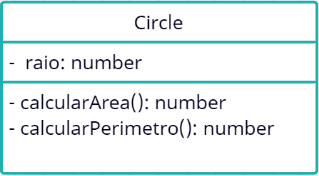
console.log("debitos:", situacaoFinanceira.valorDebitos);

console.log("saldo:", situacaoFinanceira.saldo());

**10. Represente as classes das questões 8 e 9 no formato UML. Pesquise uma**

**ferramenta on line.**

Questão 8:



Questão 9:

