

 <p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PIAUI</p>	<p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ</p> <p>Curso: ADS</p> <p>Disciplina: Programação Orientada a Objetos</p> <p>Professor: Ely</p>
--	--

Exercício 01 - extra

1. Explique com suas palavras a diferença entre uma classe e um objeto e como esses dois conceitos se relacionam. Além disso, apresente um exemplo do mundo real para cada um (por exemplo, usando a classe Carro e instâncias como Fusca e Civic).

R: Uma classe é um modelo para a criação de objetos. É ela que define as características (atributos) e os comportamentos(métodos) que um tipo de objeto vai ter, porém ela não é o objeto em si. A classe pode ser comparada a planta de uma casa, a qual especifica a quantidade de cômodos e o tipo de material a ser utilizado para a construção de uma casa, ou seja, é essencialmente o projeto da casa e não a sua forma física. Já um objeto é a instância concreta de uma classe, ou seja, é a realização do projeto estabelecido na classe, um elemento que realmente existe e que possui valores específicos para seus atributos. Tomando como base o exemplo da casa, o objeto seria a casa que foi construída através da planta. Por exemplo, a classe Carro é o modelo, enquanto um Fusca azul de 1968 seria um objeto.

2. De forma breve, conceitue atributos e métodos. Pesquise e exemplifique um objetos ou classes que possuam atributos e métodos. Por exemplo um Carro:
 - Carro
 - Atributos: cor, marca, ano
 - Métodos: acelerar(), frear()

R: Atributos são as características que definem o estado de um objeto, eles podem ser entendidos como “substantivos” na Programação Orientada a Objeto, ou seja, são as propriedades que o objeto possui. Já os métodos são as ações que um objeto pode realizar, eles podem ser entendidos como “verbos” na Programação Orientada a Objeto, os quais podem alterar os atributos de um objeto.

Exemplo: ContaBancaria

- Atributos: saldo, titular
- Métodos: depositar(), sacar()

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:

R: Sim, a importância de um atributo de uma pessoa pode variar dependendo do contexto do sistema. Já que a abstração na Programação Orientada a Objetos foca somente nos elementos essenciais para um sistema, ignorando aspectos que são irrelevantes para a resolução do problema.

Atributo	Sistema em que não é importante	Sistema em que é moderadamente importante	Sistema em que é essencial
CPF	Sistema de controle de acesso a um estacionamento	Sistema de agendamento de consultas médicas	Sistema bancário
Histórico de saúde	Sistema de rede social	Sistema de monitoramento de atividades físicas	Sistema de prontuário eletrônico de um hospital
Quantidade de seguidores	Sistema de gestão acadêmica	Sistema de e-commerce	Sistema de gerenciamento de influenciadores digitais
Habilidade destra	Sistema de controle de ponto	Sistema de videogame esportivo	Sistema de registro médico para cirurgião
Endereço	Sistema de reconhecimento facial	Sistema de streaming de vídeo	Sistema de entregas
Saldo em conta	Sistema de fórum de discussões online	Sistema de reservas de hotel	Sistema de controle financeiro
Etnia	Sistema de reservas de passagens aéreas	Sistema de análise demográfica	Sistema de pesquisa genética ou antropológica

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?

R: Sim, seria interessante em um sistema bancário um objeto Conta possuir uma Pessoa como atributo. Visto que isso representaria que cada conta pertence a uma pessoa em específico, o que representa uma forma de organização do sistema e uma garantia de que as operações financeiras estarão associadas ao titular correto.

- 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?

R: Sim, seria interessante um objeto Pessoa possuir mais de uma Conta como atributo. Visto que isso representaria o conjunto de contas de uma pessoa, com isso em mente, tem-se que o elemento da programação estrutura que melhor representaria o conjunto de contas de uma pessoa seria um array ou um conjunto, pois esses elementos permitem que uma pessoa tenha múltiplas contas associadas a ele de forma organizada.

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

Professor, disciplina, turma, coordenador, sala.

R: Curso, matrícula, histórico escolar, calendário acadêmico, secretaria.

6. Imagine um jogo qualquer. Identifique o máximo de objetos possíveis e eventuais características (atributos) e comportamentos (métodos) que eles poderiam ter.

Jogo de vôlei. Atributos: duração, local, tipo, quantidade de jogadores, placar, time1, time2. Métodos: iniciar partida, pausar, adicionar jogador, remover jogador, finalizar partidas.

R: Jogo de basquete.

Objeto: Jogo; Atributos: duracao, local, placar, time1, time2, periodoAtual;

Métodos: iniciarPartida, pausar, reiniciar, finalizarPartida, mudarPeriodo.

Objeto: Jogador; Atributos: nome, numeroDaCamisa, posicao, altura, velocidade, energia; Métodos: driblar, arremessar, passar, bloquear, rebote, correr.

Objeto: Time; Atributos: nomeDoTime, jogadores, pontos, faltas; Métodos: marcarPonto, adicionarJogador, removerJogador, marcarFalta.

Objeto: Bola; Atributos: peso, cor, circunferencia, posicaoX, posicaoY, posicaoZ; Métodos: quicar, arremessar, passar, rolar.

Objeto: Quadra; Atributos: comprimento, largura, tamanhoDaTabela, linhaDeTresPontos, linhaDeLanceLivre; Métodos: demarcarArea.

Objeto: Árbitro; Atributos: nome, posicao; Métodos: apitar, marcarFalta, pausarJogo.

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. Teste os métodos do retângulo.

R: Resposta no arquivo externo (q7.ts).

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.

R: Resposta no arquivo externo (q8.ts).

9. Crie uma classe chamada SituacaoFinanceira com os atributos valorCreditos e valorDebitos. Crie um método chamado calcularSaldo() que retorna/calcula a diferença entre crédito e débito. Instancie uma classe SituacaoFinanceira, inicialize os dois atributos e exiba o resultado do método calcularSaldo().

R: Resposta no arquivo externo (q9.ts).

10. Pesquise e apresente as classes das questões 8 e 9 no formato UML. Pesquise uma ferramenta como draw.io ou [PlantUML](https://plantuml.com)

R: Resposta nos arquivos externos (diagrama_circulo.png e plant_circulo.txt) para a representação da questão 8 e nos arquivos externos (diagrama_situacao.png e plant_situacao_fin.txt) para a representação da questão 9.