

EXERCÍCIO - LINGUAGEM DE PROGRAMAÇÃO 2 (4,0PTS)

TCG (Trading Card Games) são jogos de cartas onde jogadores elaboram estratégias combinando suas próprias cartas. Observe um exemplo de carta do jogo Yu-Gi-Oh!



Durante uma partida de YuGiOh!, uma carta qualquer pode ser composta por algumas características:

- Nome (texto);
- Número de Estrelas (número inteiro);
- Tipo (texto);
- Descrição (texto);
- Ataque (número inteiro);
- Defesa (número inteiro);
- Modo de Defesa (verdadeiro ou falso);

Sabendo disso, faça o que é pedido:

- Implemente uma classe Carta. Insira 7 atributos nesta classe de acordo com as características apresentadas.
 - Implemente um método construtor para esta classe. Este construtor irá receber 7 parâmetros e irá atribuí-los aos atributos da classe adequadamente.
- c) Implemente o método de assinatura

```
public static string CompararEstrelas(Carta c1, Carta c2)
```

Entre as duas cartas passadas como parâmetros, este método deve retornar o nome da carta que tem o maior número de estrelas. Caso as cartas tenham o mesmo número de estrelas, retorne "Mesmo nível".

- d) (LEIA COM ATENÇÃO) Implemente o método de assinatura

```
public int Atacar(Carta cartaAlvo)
```

Este método deve retornar um número inteiro indicando o dano que a carta atacante dará (número positivo) ou receberá (número negativo) na/da carta alvo passada como parâmetro.

Quando uma carta ataca uma outra carta, o dano dado ou recebido é calculado da seguinte forma:

- Se a carta alvo não está em modo de defesa:
 - Dano = Ataque da carta atacante – Ataque da carta alvo.
- Se a carta alvo está no modo de defesa:
 - Dano = Ataque da carta atacante – Defesa da carta alvo. Se Dano for maior do que zero, Dano = 0.

- e) Com base na classe implementada na questão anterior, um programador decidiu testá-la no método Main. O código que foi escrito é apresentado abaixo:
-

```
1. Carta carta_1 = new Carta("Mago Negro", 7,
                             "Mago", "O mago definitivo em termos de ataque e
                             defesa.", 2500, 2100, false);

2. Carta carta_2 = new Carta("Dragão Branco de Olhos Azuis", 8,
                             "Dragão", "Este dragão lendário é uma poderosa
                             máquina de destruição. Praticamente invencível, muito
                             poucos enfrentaram esta magnífica criatura e viveram
                             para contar a história.",
                             3000, 2500, false);

3. Carta carta_3 = new Carta("Dragão Negro de Olhos Vermelhos", 7,
                             "Dragão", "Um dragão feroz com um ataque letal.",
                             2400, 2000, false);

4. Carta carta_4 = new Carta("Lady Harpia", 4,
                             "Besta Alada", "Este animal alado com forma humana,
                             embora agrade aos olhos, é absolutamente letal em
                             batalha.",
                             1300, 1400, true);

5. Carta carta_5 = new Carta("Rei Caveira", 6,
                             "Monstro", "Um monstro com poderes das trevas para
                             confundir os inimigos.",
                             2500, 1200, true);

6. Console.WriteLine($"Entre o monstro {carta_3.Nome} e {carta_5.Nome}, o que tem
    maior número de estrelas é: ");
7. Console.WriteLine(Carta.CompararEstrelas(cartas_3, cartas_5));

8. Console.WriteLine($"{carta_2.Nome} ataca {carta_1.Nome}");
9. Console.WriteLine($"Dano calculado: ");
10. Console.WriteLine(cartas_2.Atacar(cartas_1));

11. Console.WriteLine($"{cartas_1.Nome} ataca {cartas_4.Nome}");
12. Console.WriteLine($"Dano calculado: ");
13. Console.WriteLine(cartas_1.Atacar(cartas_4));
```

Responda o que é pedido:

- Quantos objetos da classe Carta foram criados?
- O que será mostrado no Console quando a linha 7 for executada?
- O que será mostrado no Console quando a linha 10 for executada?
- O que será mostrado no Console quando a linha 13 for executada?