

**1ª CERTIFICAÇÃO - (LINGUAGEM DE PROGRAMAÇÃO 2)  
DRAGON BALL "SERIES"**

NOME: \_\_\_\_\_ NÚMERO: \_\_\_\_\_ TURMA: DS \_\_\_\_\_ DATA: 17/10/2024

**CONTEXTUALIZAÇÃO**

Dragon Ball é uma icônica franquia de anime e mangá criada por Akira Toriyama, que acompanha as aventuras de Goku e seus amigos em busca das lendárias esferas do dragão, enfrentando inimigos poderosos ao longo do caminho. Recentemente, a série ganhou uma nova expansão com a estreia de Dragon Ball Daima, além do lançamento do jogo Dragon Ball Sparking! ZERO, título desenvolvido na Spike Chunsoft e publicado pela Bandai Namco. Com apenas algumas horas de lançamento, o game superou a marca de 122 mil jogadores simultâneos na plataforma Steam.

**QUESTÃO 01 – GUERREIROS "Z" (1,0 PT)**

O nome **Guerreiros Z** foi inspirado pela fase conhecida como **Dragon Ball Z** e reforça a importância da letra "Z" como um símbolo de uma nova era de batalhas épicas. Eles são os heróis principais dessa fase, protegendo a Terra contra ameaças maiores e mais poderosas do que antes.

Implemente um programa em C# (Console) que desenhe a letra "Z" na tela de acordo com a altura fornecida pelo usuário. O programa deverá funcionar da seguinte forma:

- O usuário deve informar um número inteiro representando a altura do "Z". A altura mínima permitida é 3.
- O programa deverá desenhar a letra "Z" de forma proporcional à altura fornecida, utilizando o caractere Z para representar a letra.

**Exemplo de funcionamento:**

Altura: 2	Altura: 3	Altura: 4	Altura: 5
Altura inválida.	ZZZ	ZZZZ	ZZZZZ
	Z	Z	Z
	ZZZ	Z	Z
		ZZZZ	Z
			ZZZZZ

Exemplos de Entradas e Saídas



## QUESTÃO 02 – SPARKING! ZERO (2,0 PTS)

*Dragon Ball Sparking! ZERO é um jogo de luta recém-lançado que conta com mais de 180 personagens jogáveis, muitos dos quais possuem transformações que, do ponto de vista da programação, são tratadas como outros personagens. Essas transformações ampliam as possibilidades estratégicas no combate, trazendo uma dinâmica fiel ao universo de Dragon Ball.*

À luz dos conceitos de classes e objetos, faça o que é pedido:

- Crie a classe **Personagem**. Essa classe contém três atributos: um **Nome** (valor textual), um **Poder de Luta** (valor inteiro) e uma **Transformação** (um objeto da classe personagem). (0,5 pt)
- Implemente um **método construtor** para esta classe. Este método irá receber apenas 3 parâmetros (nome, poder de luta e transformação) e irá atribuí-los aos atributos da classe. (0,5 pt)
- Implemente o método de assinatura

```
public int NumeroDeTransformacoes()
```

na classe **Personagem**. Este método retorna um número inteiro indicando quantas transformações esse personagem tem no total, considerando a sua **cadeia de transformações**. (1,0 pt)

**Definição:** uma **cadeia de transformações** é uma sequência de evoluções pelas quais um personagem pode passar. Um personagem pode ter uma transformação associada, representada pelo atributo transformação. Se o atributo transformação for diferente de null, isso significa que o personagem pode se transformar em outro personagem. Essa lógica se aplica recursivamente: se o próximo personagem na sequência também tiver uma transformação, ele poderá se transformar novamente, e assim por diante.

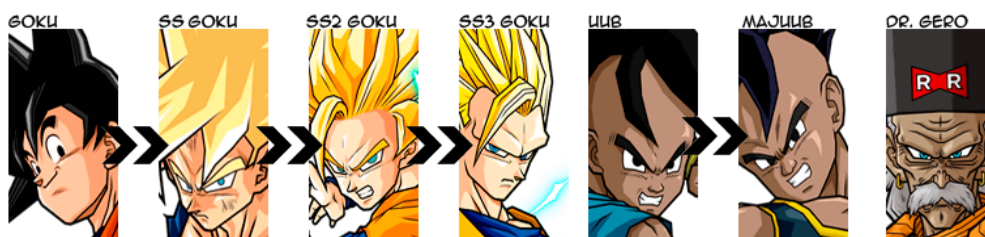
**Exemplo:** considere os seguintes objetos assim como as suas transformações:

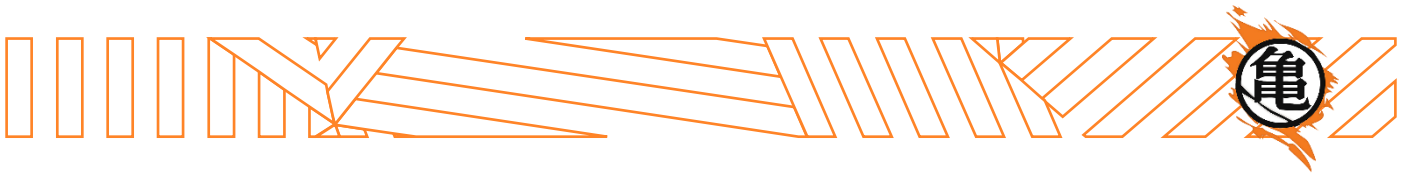
```
Personagem goku_ssj3 = new Personagem("SS3 Goku", 250000, null);
Personagem goku_ssj2 = new Personagem("SS2 Goku", 100000, goku_ssj3);
Personagem goku_ssj = new Personagem("SS Goku", 50000, goku_ssj2);
Personagem goku = new Personagem("Goku", 10000, goku_ssj);

Personagem majuub = new Personagem("Majuub", 245000, null);
Personagem uub = new Personagem("Uub", 140000, majuub);

Personagem dr_gero = new Personagem("Dr. Gero", 45000, null);
//-----
Console.WriteLine(goku.NumeroDeTransformacoes()); // Saída 3
Console.WriteLine(uub.NumeroDeTransformacoes()); // Saída 1
Console.WriteLine(dr_gero.NumeroDeTransformacoes()); // Saída 0
```

Aqui, o personagem “Goku” tem três transformações (“SS Goku”, “SS2 Goku” e “SS3 Goku”), o personagem “Uub” tem uma transformação (“Majuub”) e o personagem “Dr. Gero” não tem nenhuma transformação.






## QUESTÃO 03 – THE DRAGON BALLS (2,0 PTS)

As **Dragon Balls** são esferas mágicas com estrelas numeradas de um a sete, que, quando reunidas, invocam o dragão Shenlong para conceder qualquer desejo.

Considere os dois arquivos, conforme é mostrado abaixo, pertencentes a um mesmo projeto.

DragonBall.cs	Program.cs
<pre>class DragonBall {     private int stars = 1;     public DragonBall(int stars)     {         SetStars(stars);     }      public void SetStars(int stars)     {         if (stars &gt;= 1 &amp;&amp; stars &lt;= 7)             this.stars = stars;     }      public int GetStars()     {         return this.stars;     }      public string Descrever()     {         string texto = "";          for (int i = 0; i &lt; this.stars; i++)             texto += "*";          return texto;     } }</pre>	<pre>List&lt;DragonBall&gt; dragonBalls = new List&lt;DragonBall&gt;();  dragonBalls.Add(new DragonBall(1)); dragonBalls.Add(new DragonBall(2)); dragonBalls.Add(new DragonBall(3)); dragonBalls.Add(new DragonBall(4)); dragonBalls.Add(new DragonBall(5)); dragonBalls.Add(new DragonBall(6)); dragonBalls.Add(new DragonBall(7));  Console.WriteLine(dragonBalls[4].Descrever());  if (DragonBall.PodeFazerDesejo(dragonBalls))     Console.WriteLine("Desejo passar em LP2!"); else     Console.WriteLine("Você não juntou todas as esferas do dragão...");</pre> 

a) Observe o código escrito no arquivo Program.cs. e indique o que será impresso no Console de acordo com a linha marcada de amarelo. (0,5 pt)

b) Suponha que no arquivo Program.cs, após a linha marcada de amarelo, a instrução

```
Console.WriteLine(dragonBalls[4].stars);
```

seja inserida. O código será compilado e executado? Justifique a sua resposta. (0,5 pt)

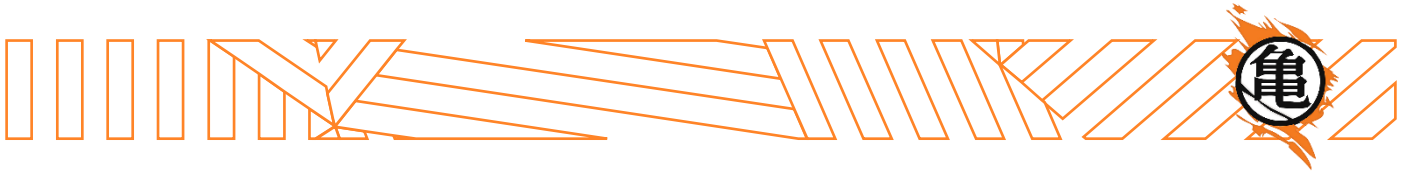
c) Implemente a função de assinatura

```
public static bool PodeFazerDesejo(List<DragonBall> dragonBalls).
```

Esta função recebe uma lista de objetos do tipo DragonBall (de qualquer tamanho) e retorna um valor booleano caso seja possível **realizar um desejo**.

**Definição:** para **realizar um desejo** é necessário verificar se a lista contém pelo menos um objeto com 1, 2, 3, 4, 5, 6 e 7 estrelas (stars). (1,0 pt)





RASCUNHO

Boa Prova!

