



Instituto Infnet

Fundamentos de Desenvolvimento com C#

Desenvolvimento Back-end

2025T2

Lohran Fellipe Mendes de Souza

TP3 - Escrever programas em C# que utilizem classes e objetos

Observação:

O tp e os enunciados das questões parecem ter sido criados por uma inteligência artificial, pedindo para que a gente entregasse as respostas de uma maneira pouco natural, mesmo assim, eu tentei fazer tudo o que foi pedido ao pé da letra, por isso verá que no código há partes que não há nenhuma saída no console, pois a questão não pedia, e outras partes que não estão sendo instanciadas no método Main, fiz isso para facilitar o entendimento e porque a maneira que a questão pediu dificulta a compreensão do resto do código, qualquer comentário pode me retornar.

Exercício 1

A linguagem de programação C# usa um paradigma de programação orientada a objetos, objetos que são criados a partir de classes.

Tá bom, mas o que são Classes?

Classes são criadas em C# para definir a estrutura ou um padrão de objetos com que iremos trabalhar, dentro dessas classes nós definimos a regra de negócio, seus métodos e atributos, feito isso, nós podemos instâncias desses objetos e trabalhar em cima de suas funcionalidades.

Atributos são as características desse objeto, como a cor dele, o peso, altura, estas são propriedades necessárias que podem ou devem ser usadas dentro do código, enquanto os métodos são ações que esse objeto criado a partir de uma classe deve fazer.

3

Como exemplo, criei uma classe de carros, e irei instanciar dois diferentes:

```
C# Program.cs    C# Carro.cs ×
1 {} namespace lohran_mendes_DR3_TP3;
2
3 public class Carro
4 {
5     public string Nome;
6     public string Marca;
7
8     public Carro(string nome, string marca)
9     {
10         Nome = nome;
11         Marca = marca;
12     }
13
14     private string Buzinar() {
15         return "0 " + Nome + " da marca " + Marca + " está buzinando!";
16     }
17
18     // Exemplo da criação de objeto a partir da classe declarada
19     public static void Exercicio_1(){
20         Console.WriteLine("\nExercício 1:");
21         var carro1 = new Carro( nome: "HB20", marca: "Hyundai");
22         var carro2 = new Carro( nome: "Virtus", marca: "Volkswagen");
23         Console.WriteLine(carro1.Buzinar());
24         Console.WriteLine(carro2.Buzinar());
25     }
26 }
```

```
1 {} namespace lohran_mendes_DR3_TP3;
2
3 class Program
4 {
5     static void Main(string[] args)
6     {
7         Carro.Exercicio_1();
8     }
9 }
```

Run lohran_mendes_DR3_TP3

Exercício 1:
0 HB20 da marca Hyundai está buzinando!
0 Virtus da marca Volkswagen está buzinando!
Process finished with exit code 0.

Como no exemplo mostrado, os objetos carro1 e carro2 foram criados, cada um com suas próprias propriedades de nome e marca, e os dois tinham um método que corresponde a uma ação que todo objeto da classe carro deve ter, o de buzinar.

Exercício 2

Cada atributo é importante pois são essenciais no contexto apresentado de uma venda de ingressos para shows, todo ingresso precisa ter um nome do show para identificá-lo, um preço para que esse seja vendido, e a sua

5

quantidade disponível no estoque para que o vendedor consiga fazer o controle.

```
C# Program.cs  C# Ingresso.cs x
1 {} namespace lohran_mendes_DR3_TP3;
2
3 public class Ingresso
4 {
5     string NomeDoShow;
6     double Preco;
7     int QuantidadeDisponivel;
8 }
```

Exercício 3

```
C# Program.cs  C# Ingresso.cs x
1 {} namespace lohran_mendes_DR3_TP3;
2
3 public class Ingresso
4 {
5     string NomeDoShow;
6     double Preco;
7     int QuantidadeDisponivel;
8
9     public void AlterarPreco(double novoPreco)
10    {
11        Preco = novoPreco;
12    }
13
14    public void AlterarQuantidade(int novaQuantidade)
15    {
16        QuantidadeDisponivel = novaQuantidade;
17    }
18
19    public string ExibirInformacoes()
20    {
21        return $"0 ingresso para {NomeDoShow} está custando R${Preco} e ainda temos {QuantidadeDisponivel} disponiveis.";
22    }
23 }
24
```

Exercício 4

```
1 usage
24 > public static void Exercicio_4()
25 {
26     Console.WriteLine("\nExercicio 4:");
27     Ingresso ingresso = new();
28     ingresso.NomeDoShow = "Thiaguinho";
29     ingresso.Preco = 80.50;
30     ingresso.QuantidadeDisponivel = 100;
31     Console.WriteLine("Informações do ingresso antes de altera-lo:");
32     Console.WriteLine(ingresso.ExibirInformacoes());
33     ingresso.AlterarPreco(150);
34     ingresso.AlterarQuantidade(70);
35     Console.WriteLine("Informações do ingresso depois da alteração:");
36     Console.WriteLine(ingresso.ExibirInformacoes());
37 }
38 }
39 }
```

```
C# Program.cs x C# Ingresso.cs
1 {} namespace lohran_mendes_DR3_TP3;
2
3 class Program
4 {
5     static void Main(string[] args)
6     {
7         // Carro.Exercicio_1();
8         Ingresso.Exercicio_4();
9     }
10 }
```

Run lohran_mendes_DR3_TP3 x

Exercicio 4:
Informações do ingresso antes de altera-lo:
0 ingresso para Thiaguinho está custando R\$80,5 e ainda temos 100 disponíveis.
Informações do ingresso depois da alteração:
0 ingresso para Thiaguinho está custando R\$150 e ainda temos 70 disponíveis.

Process finished with exit code 0.

Exercício 5

Métodos de getters e setters são úteis quando queremos buscar ou alterar a propriedade de algum objeto que já foi instanciado, eles são úteis por causa da padronização de implementação, pela possibilidade de mexer em propriedades privadas e sobre a flexibilidade que ele nos fornece, permitindo que coloquemos validações quando eles forem usados, como, não permitir colocar um valor negativo no preço de um produto.

```
C# Program.cs C# Ingresso.cs x
1 namespace lohran_mendes_DR3_TP3;
2
3 public class Ingresso
4 {
5     private string NomeDoShow { get; set; } = "";
6     double Preco { get; set; }
7     int QuantidadeDisponivel { get; set; }
8 }
```

```
C# Program.cs C# Ingresso.cs x
3 public class Ingresso
38
39 public static void Exercicio_5()
40 {
41     Console.WriteLine("\nExercício 5:");
42     Ingresso ingresso = new();
43     ingresso.NomeDoShow = "Lady Gaga";
44     ingresso.Preco = 200;
45     ingresso.QuantidadeDisponivel = 400;
46     Console.WriteLine($"Nome do show através do getter: {ingresso.NomeDoShow}");
47     Console.WriteLine($"Preço do show através do getter: {ingresso.Preco}");
48     Console.WriteLine($"Quantidade disponível de ingressos do show através do getter: {ingresso.QuantidadeDisponivel}");
49 }
50
51
```

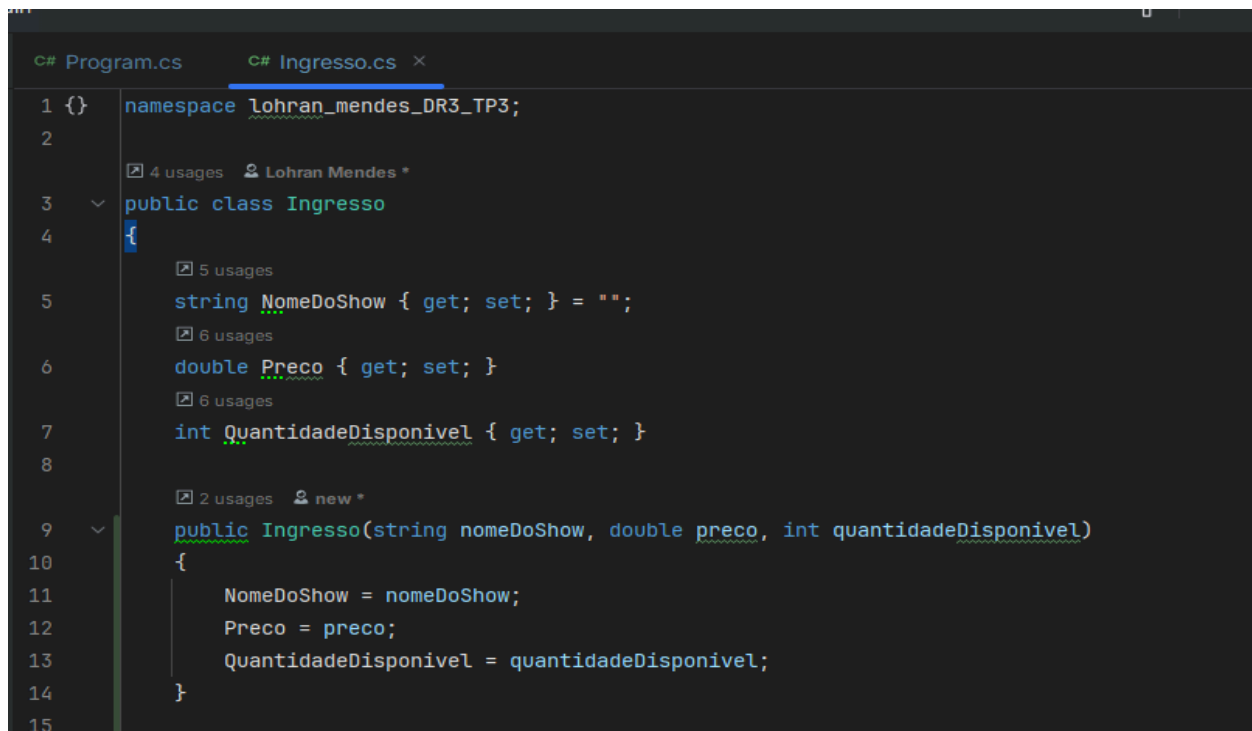
Run lohran_mendes_DR3_TP3 x

C:/Users/Asus/Desktop/INFNET/trabalhos/lohran_mendes_DR3_TP3/lohran_mendes_DR3_TP3/bin/Debug/net9.0/lohran_mendes_DR3_TP3.exe

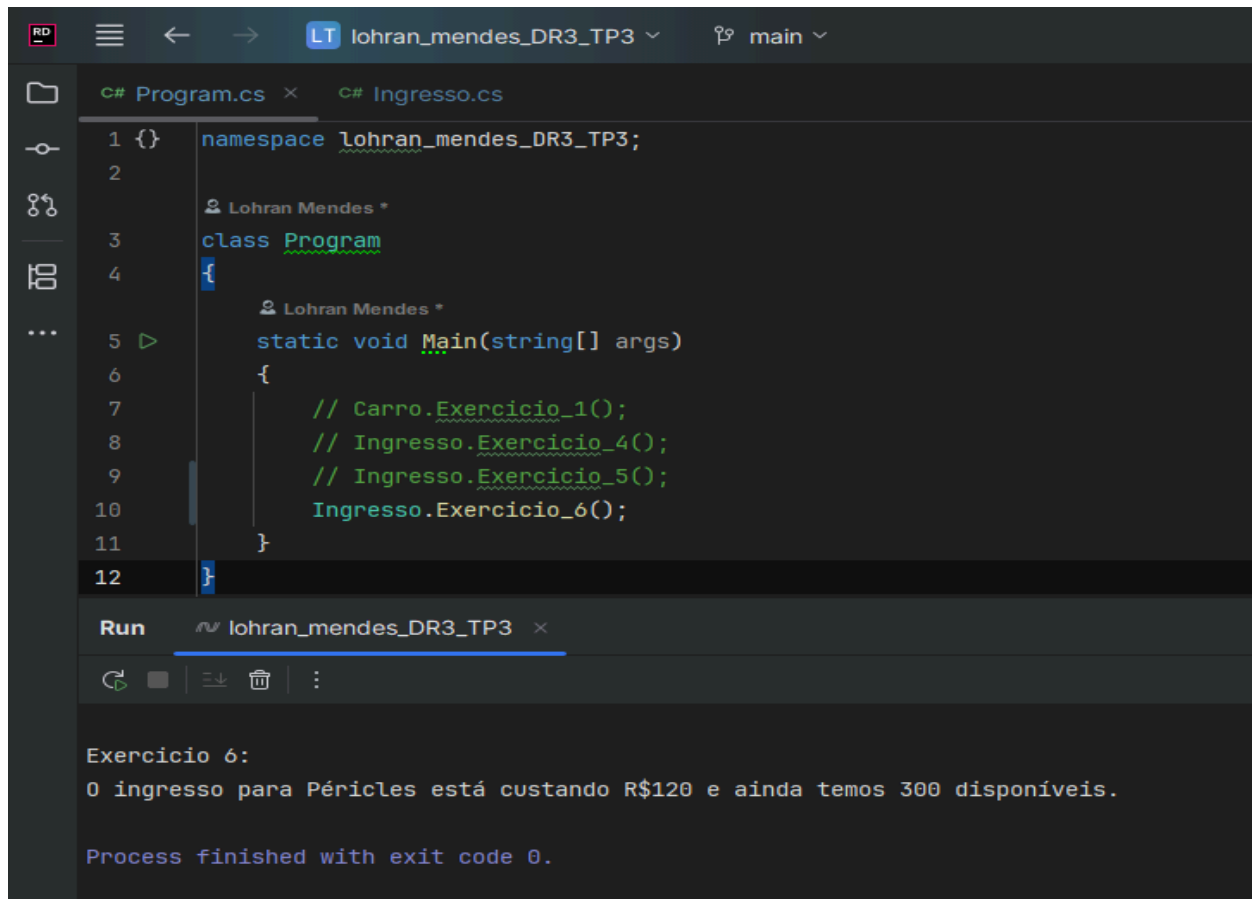
Exercício 5:
Nome do show através do getter: Lady Gaga
Preço do show através do getter: 200
Quantidade disponível de ingressos do show através do getter: 400

Exercício 6

O uso de construtores facilita a criação do objeto, fazendo a utilização de menos código para instanciá-lo, além de nos obrigar a instanciar o objeto com certas propriedades, no caso eu criei dois construtores para que o meu código dos outros exercícios continuasse funcionando, mas caso eu criasse apenas um construtor com propriedades obrigatórias, os meus outros objetos dariam erro até eu atualizar o meu construtor.



```
1 {} namespace Lohran_mendes_DR3_TP3;
2
3 public class Ingresso
4 {
5     string NomeDoShow { get; set; } = "";
6     double Preco { get; set; }
7     int QuantidadeDisponivel { get; set; }
8
9     public Ingresso(string nomeDoShow, double preco, int quantidadeDisponivel)
10    {
11        NomeDoShow = nomeDoShow;
12        Preco = preco;
13        QuantidadeDisponivel = quantidadeDisponivel;
14    }
15 }
```

The image shows a Visual Studio IDE window with a C# project named 'lohran_mendes_DR3_TP3'. The file explorer on the left shows two files: 'Program.cs' and 'Ingresso.cs'. The 'Program.cs' file is open, showing the following code:

```
1 {} namespace lohran_mendes_DR3_TP3;
2
3 class Program
4 {
5     static void Main(string[] args)
6     {
7         // Carro.Exercicio_1();
8         // Ingresso.Exercicio_4();
9         // Ingresso.Exercicio_5();
10        Ingresso.Exercicio_6();
11    }
12 }
```

Below the code editor, the 'Run' window is open, showing the output of the program:

```
Exercicio 6:
0 ingresso para Péricles está custando R$120 e ainda temos 300 disponíveis.

Process finished with exit code 0.
```

Exercício 7

```
ain v
C# Program.cs C# Matricula.cs x C# Ingresso.cs
1 {} namespace lohrran_mendes_DR3_TP3;
2
3 public class Matricula
4 {
5     string NomeDoAluno;
6     string Curso;
7     int NumeroMatricula;
8     string Situacao;
9     string DataInicial;
10 }
```

Exercício 8

```
C# Program.cs C# Matricula.cs x C# Ingresso.cs
1 {} namespace lohrran_mendes_DR3_TP3;
2
3 public class Matricula
4 {
5     string NomeDoAluno;
6     string Curso;
7     int NumeroMatricula;
8     string Situacao;
9     string DataInicial;
10
11     public void Trancar()
12     {
13         Situacao = "Trancada";
14     }
15
16     public void Reativar()
17     {
18         Situacao = "Ativa";
19     }
20
21     public string ExibirInformacoes()
22     {
23         return $"*0 aluno {NomeDoAluno} do curso {Curso} iniciou a formação em {DataInicial} e está com a matricula {Situacao}.";
24     }
25 }
```

Exercício 9

```

25
26 1usage
27 public static void Exercicio_9()
28 {
29     Console.WriteLine("\nExercício 9:");
30     Matricula matricula = new Matricula();
31     matricula.NomeDoAluno = "Lohran Mendes";
32     matricula.Curso = "Engenharia de Software";
33     matricula.DataInicial = "01/09/2024";
34     matricula.Trancar();
35     Console.WriteLine("Informações do aluno antes das alterações:");
36     Console.WriteLine(matricula.ExibirInformacoes());
37     matricula.Reativar();
38     Console.WriteLine("Informações do aluno depois das alterações:");
39     Console.WriteLine(matricula.ExibirInformacoes());
40
41 }
42

```

```

3 class Program
4 {
5     static void Main(string[] args)
6     {
7         // Carro.Exercicio_1();
8         // Ingresso.Exercicio_4();
9         // Ingresso.Exercicio_5();
10        // Ingresso.Exercicio_6();
11        Matricula.Exercicio_9();
12    }

```

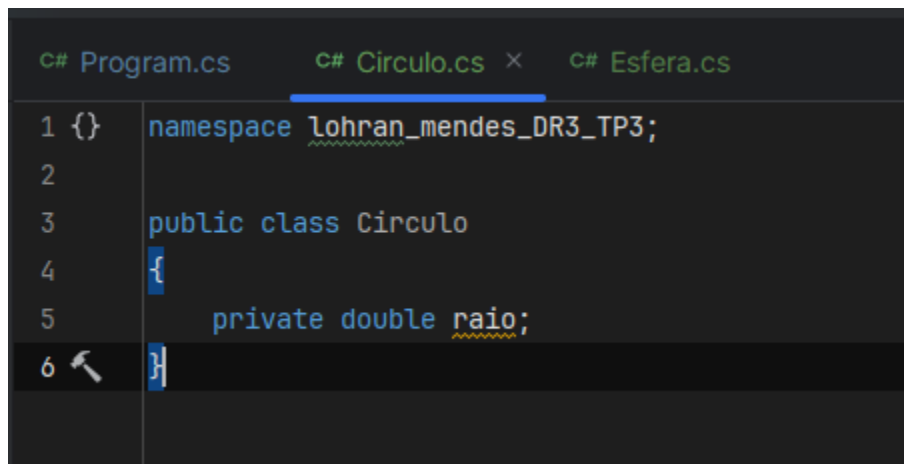
Run lohnan_mendes_DR3_TP3

Exercício 9:
 Informações do aluno antes das alterações:
 O aluno Lohran Mendes do curso Engenharia de Software iniciou a formação em 01/09/2024 e está com a matrícula Trancada.
 Informações do aluno depois das alterações:
 O aluno Lohran Mendes do curso Engenharia de Software iniciou a formação em 01/09/2024 e está com a matrícula Ativa.

Process finished with exit code 0.

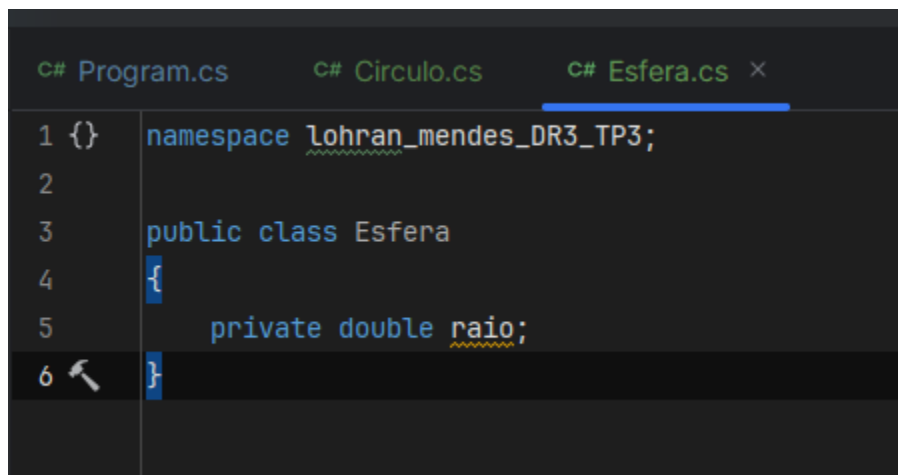
Exercício 10

O atributo raio é importante tanto para a classe circulo quanto a esfera porque é a propriedade que define essas formas, sem a qual elas não podem existir matematicamente. Além disso, qualquer tipo de método ou calculo adicionado posteriormente precisa do raio.



The screenshot shows a code editor with three tabs: 'C# Program.cs', 'C# Circulo.cs' (selected), and 'C# Esfera.cs'. The code in the 'C# Circulo.cs' tab is as follows:

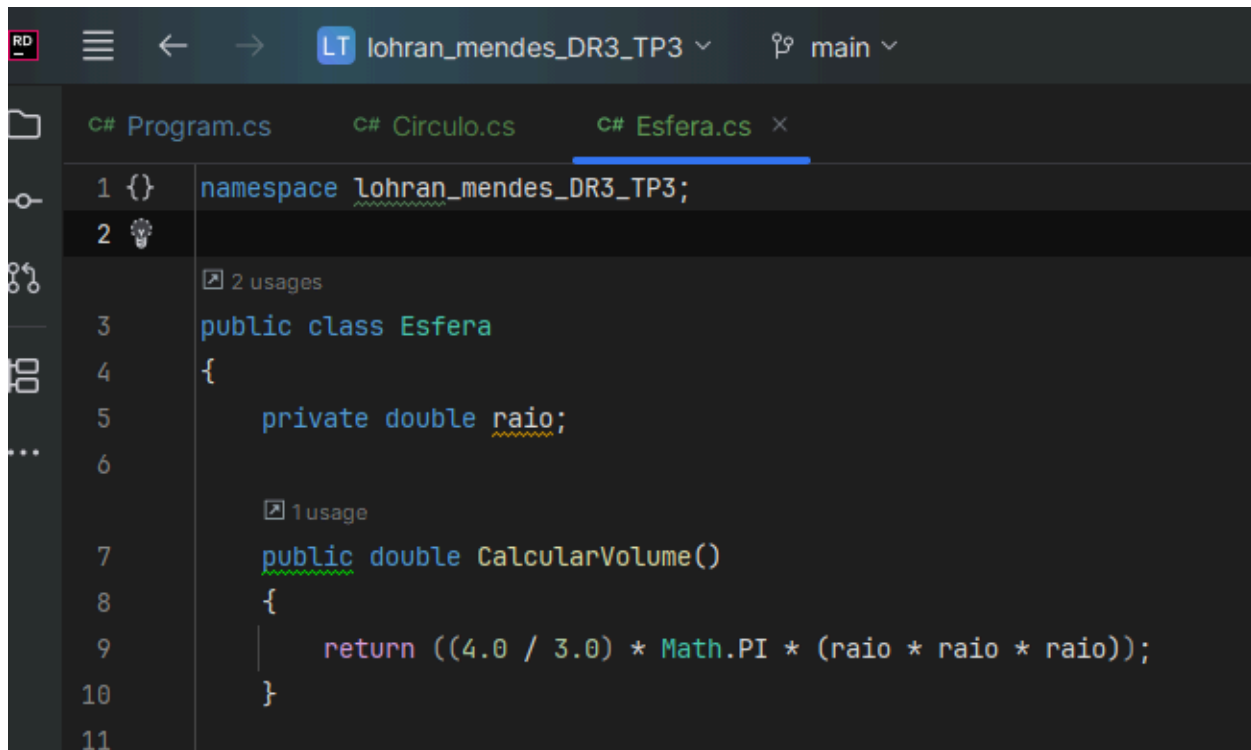
```
1 {} namespace lohran_mendes_DR3_TP3;  
2  
3 public class Circulo  
4 {  
5     private double raio;  
6 }
```



The screenshot shows a code editor with three tabs: 'C# Program.cs', 'C# Circulo.cs', and 'C# Esfera.cs' (selected). The code in the 'C# Esfera.cs' tab is as follows:

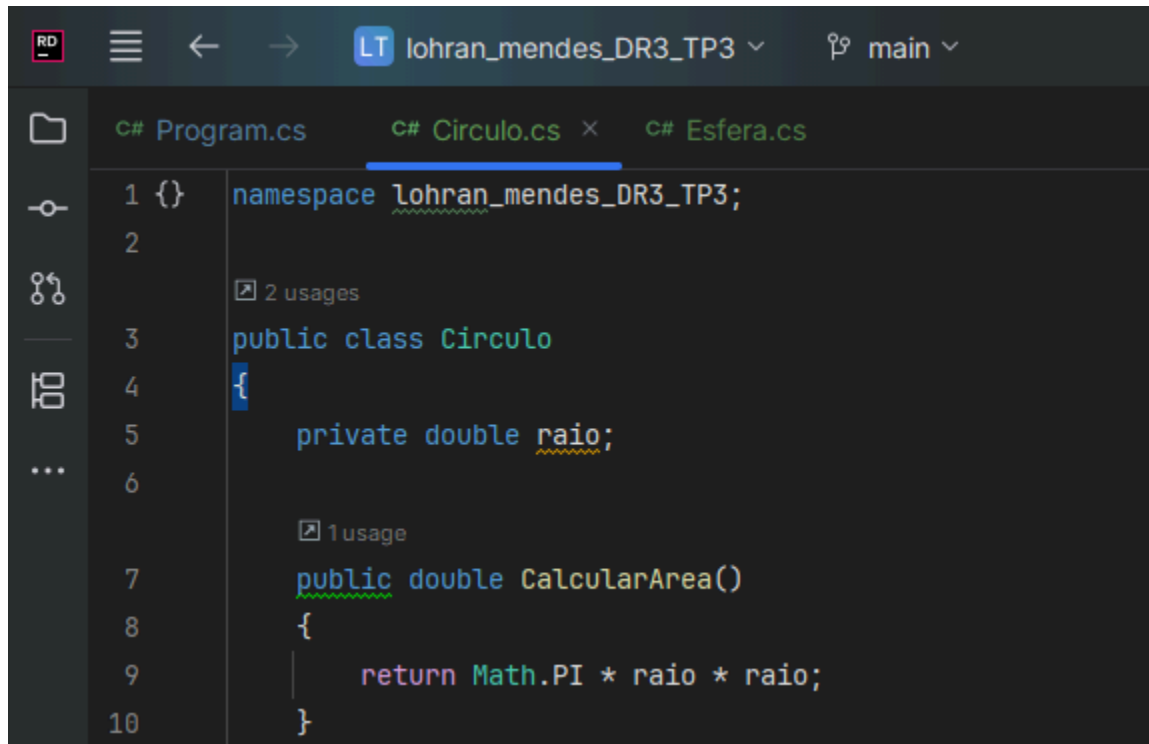
```
1 {} namespace lohran_mendes_DR3_TP3;  
2  
3 public class Esfera  
4 {  
5     private double raio;  
6 }
```

Exercício 11



The screenshot shows the Visual Studio IDE with the file explorer on the left and the code editor in the center. The active file is `Esfera.cs` in the `lohnan_mendes_DR3_TP3` project. The code defines a `namespace lohan_mendes_DR3_TP3;` and a `public class Esfera`. Inside the class, there is a `private double raio;` and a `public double CalcularVolume()` method. The method body returns the volume of a sphere using the formula $\frac{4.0}{3.0} * \text{Math.PI} * (\text{raio} * \text{raio} * \text{raio})$.

```
1 {} namespace lohan_mendes_DR3_TP3;
2
3 public class Esfera
4 {
5     private double raio;
6
7     public double CalcularVolume()
8     {
9         return ((4.0 / 3.0) * Math.PI * (raio * raio * raio));
10    }
11
```

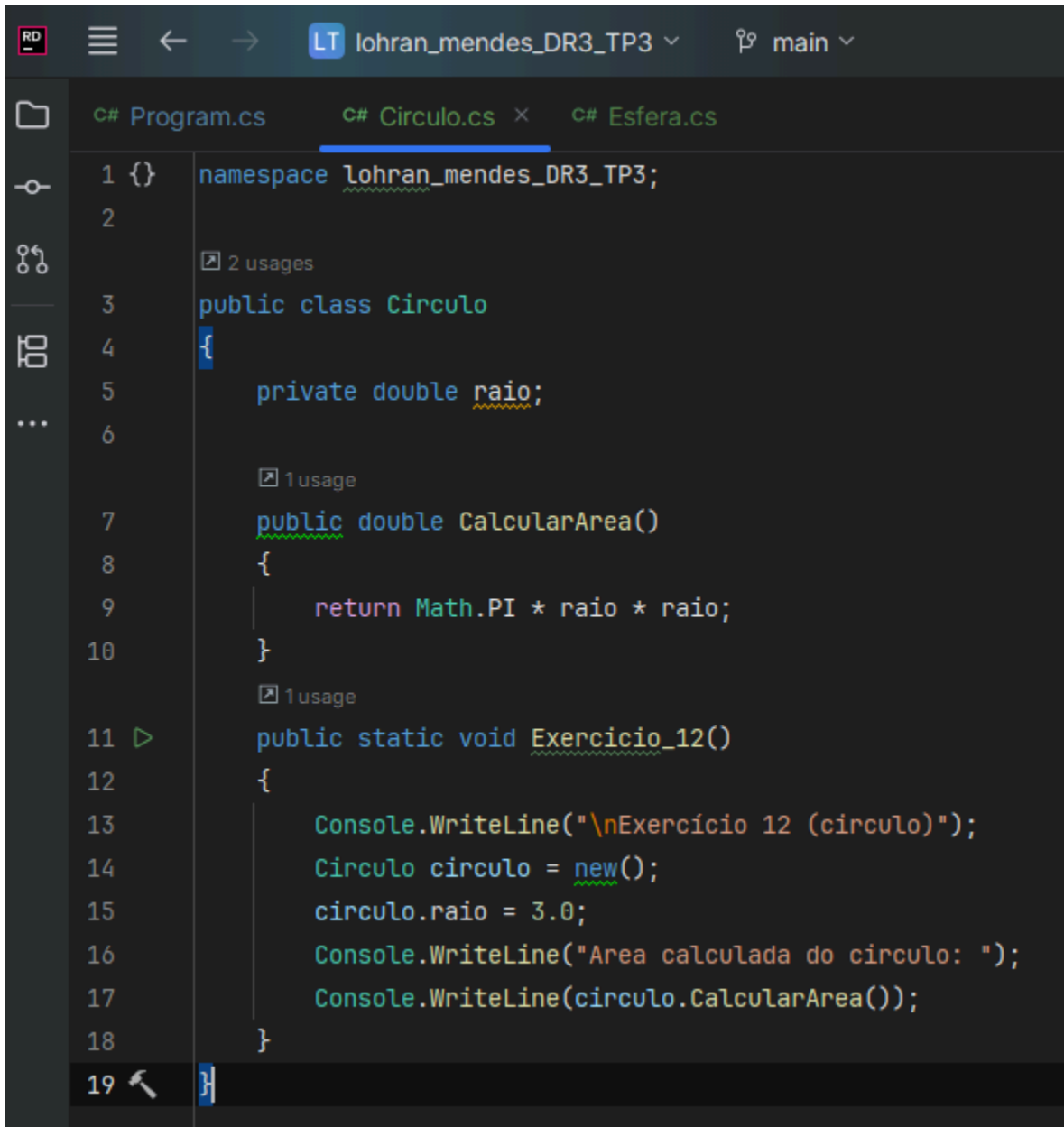


The screenshot shows the Visual Studio IDE with the file explorer on the left and the code editor in the center. The active file is `Circulo.cs` in the `lohnan_mendes_DR3_TP3` project. The code defines a `namespace lohan_mendes_DR3_TP3;` and a `public class Circulo`. Inside the class, there is a `private double raio;` and a `public double CalcularArea()` method. The method body returns the area of a circle using the formula $\text{Math.PI} * \text{raio} * \text{raio}$.

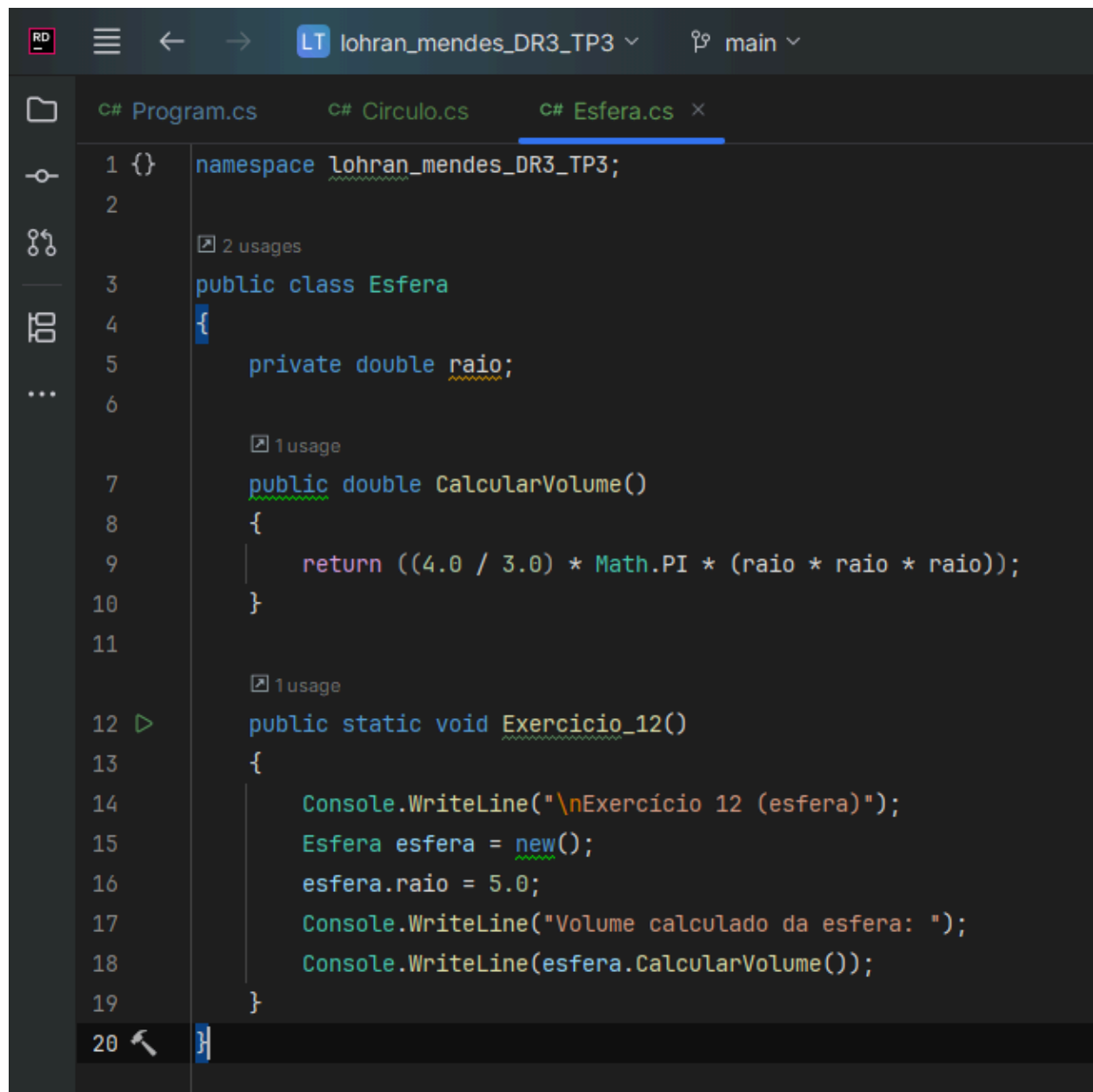
```
1 {} namespace lohan_mendes_DR3_TP3;
2
3 public class Circulo
4 {
5     private double raio;
6
7     public double CalcularArea()
8     {
9         return Math.PI * raio * raio;
10    }

```

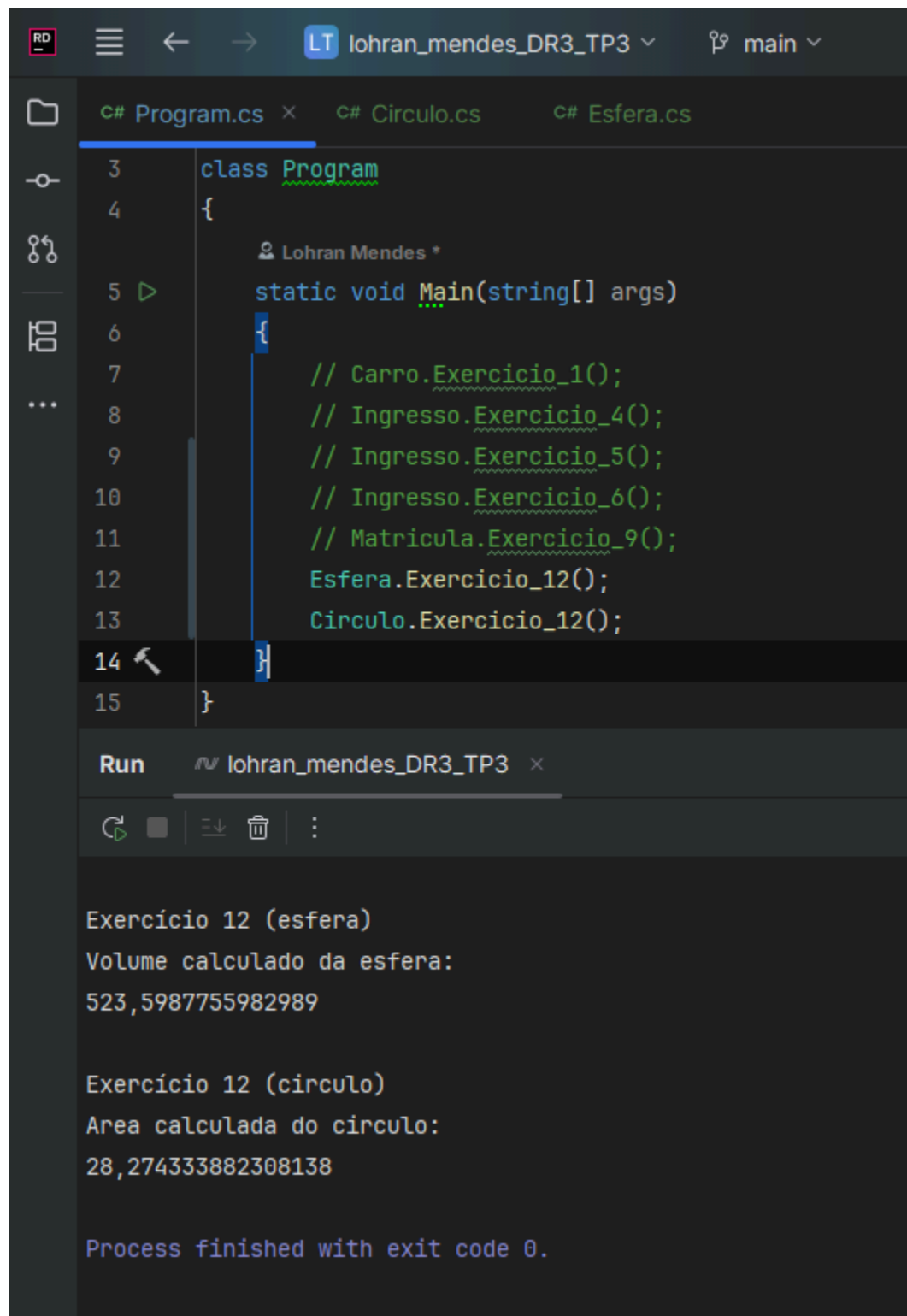
Exercício 12



```
1 {} namespace lohran_mendes_DR3_TP3;  
2  
3 public class Circulo  
4 {  
5     private double raio;  
6  
7     public double CalcularArea()  
8     {  
9         return Math.PI * raio * raio;  
10    }  
11  
12    public static void Exercicio_12()  
13    {  
14        Console.WriteLine("\nExercício 12 (circulo)");  
15        Circulo circulo = new();  
16        circulo.raio = 3.0;  
17        Console.WriteLine("Area calculada do circulo: ");  
18        Console.WriteLine(circulo.CalcularArea());  
19    }
```



```
1 {} namespace lohran_mendes_DR3_TP3;
2
3 public class Esfera
4 {
5     private double raio;
6
7     public double CalcularVolume()
8     {
9         return ((4.0 / 3.0) * Math.PI * (raio * raio * raio));
10    }
11
12    public static void Exercicio_12()
13    {
14        Console.WriteLine("\nExercício 12 (esfera)");
15        Esfera esfera = new();
16        esfera.raio = 5.0;
17        Console.WriteLine("Volume calculado da esfera: ");
18        Console.WriteLine(esfera.CalcularVolume());
19    }
20 }
```



The image shows a Visual Studio Code editor window with a C# project named "lohran_mendes_DR3_TP3". The editor displays the "Program.cs" file, which contains a class named "Program" with a static method "Main". The "Main" method calls several static methods from other classes: "Carro.Exercicio_1()", "Ingresso.Exercicio_4()", "Ingresso.Exercicio_5()", "Ingresso.Exercicio_6()", "Matricula.Exercicio_9()", "Esfera.Exercicio_12()", and "Circulo.Exercicio_12()". The output window at the bottom shows the results of these calls, including the volume of a sphere and the area of a circle, followed by a message indicating the process finished with exit code 0.

```
class Program
{
    static void Main(string[] args)
    {
        // Carro.Exercicio_1();
        // Ingresso.Exercicio_4();
        // Ingresso.Exercicio_5();
        // Ingresso.Exercicio_6();
        // Matricula.Exercicio_9();
        Esfera.Exercicio_12();
        Circulo.Exercicio_12();
    }
}
```

Run lohran_mendes_DR3_TP3

Exercicio 12 (esfera)
Volume calculado da esfera:
523,5987755982989

Exercicio 12 (circulo)
Area calculada do circulo:
28,274333882308138

Process finished with exit code 0.