



Interativa

Unidade II

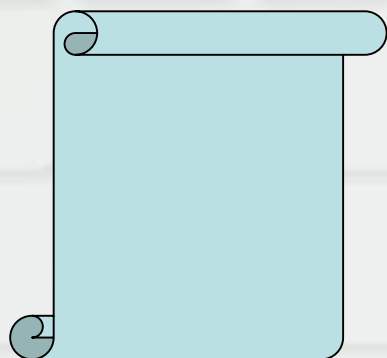
PROGRAMAÇÃO ORIENTADA A OBJETOS I

Prof. Cassiano Gunji

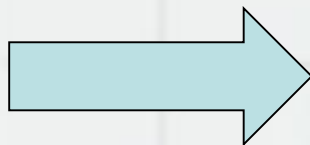


Conceitos básicos

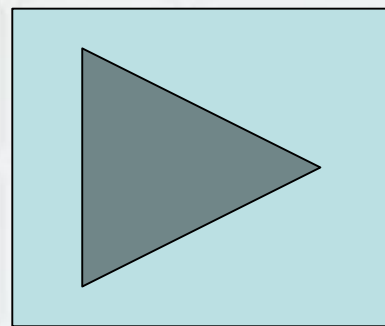
- Linguagens de programação tradicionais são compiladas diretamente para a linguagem de máquina.



Código-fonte



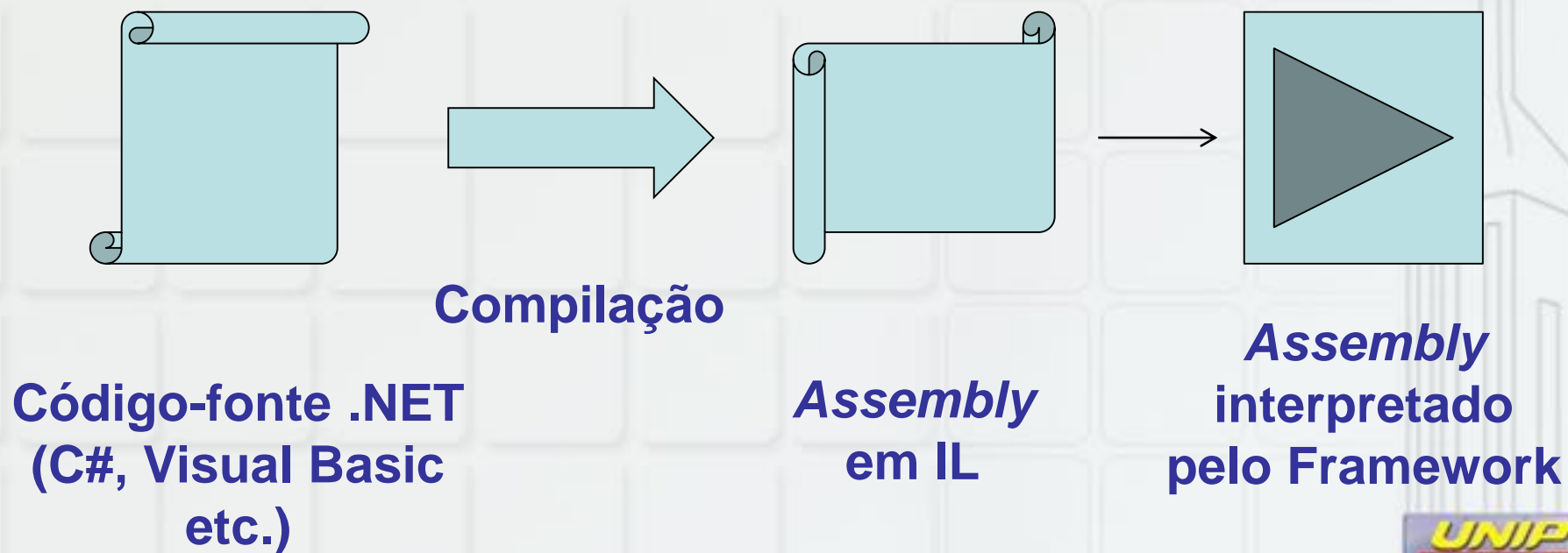
Compilação



Programa

Conceitos básicos

- Linguagens do Framework .NET são compiladas para um *assembly* que não são escritos diretamente em linguagem de máquina, e sim em *Intermediate Language* (IL – Linguagem Intermediária).



Vantagens da arquitetura do Framework .NET

- Independência da linguagem de programação.
- Reutilização de código legado.
- Tempo de execução compartilhado.
- Sistemas autoexplicativos e controle de versões.
- Simplicidade na resolução de problemas complexos.



Visual Studio .NET

O Visual Studio .NET é o ambiente de desenvolvimento integrado da Microsoft para desenvolver sistemas usando o Framework .NET:

- Visual C# Express Edition;
- Visual Studio Professional;
- Visual Studio Ultimate etc.



Visual Studio .NET

- Endereço para *download* no livro-texto.
- Tutorial para criar um projeto no Visual C# Express no livro-texto.
- Tutorial para criar um projeto no Visual Studio no livro-texto.



Características da Linguagem C#

- Clareza, simplicidade e facilidade.
- É completamente orientada a objetos.
- Não requer o uso de ponteiros para gerenciar a memória.
- Suporta os conceitos mais úteis de orientação a objetos (interfaces, herança, polimorfismo etc.).
- Possui código 100% reutilizável por outras linguagens de programação do Framework .NET.

Primeiro programa (livro-texto)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Primeiro_Programa
{
    References
    class Program
    {
        /* O trecho abaixo é o método Main.
        * O método Main é o ponto por onde um
        * programa é iniciado.*/
        References
        static void Main(string[] args)
        {
            // escrevendo no console
            Console.WriteLine("Olá mundo em C#");
            Console.ReadLine();
        }
    }
}
```


Interatividade

Qual das alternativas abaixo pode ser atribuída como uma característica do Framework .NET?

- a) É um ambiente capaz de executar código semicompilado para a Linguagem Intermediária .NET.**
- b) Pode ser utilizado para criar sistemas usando as linguagens de programação .NET sem a necessidade de utilizar o Visual Studio.**
- c) Podemos criar sistemas .NET sem o Framework se estivermos usando o Visual Studio como ambiente de desenvolvimento.**
- d) Podemos executar sistemas .NET sem o Framework se o Visual Studio estiver disponível na máquina.**
- e) O Framework .NET só está disponível em sistemas que utilizem o Windows como sistema operacional.**

Resposta

Qual das alternativas abaixo pode ser atribuída como uma característica do Framework .NET?

- a) **É um ambiente capaz de executar código semicompilado para a Linguagem Intermediária .NET.**
- b) Pode ser utilizado para criar sistemas usando as linguagens de programação .NET sem a necessidade de utilizar o Visual Studio.
- c) Podemos criar sistemas .NET sem o Framework se estivermos usando o Visual Studio como ambiente de desenvolvimento.
- d) Podemos executar sistemas .NET sem o Framework se o Visual Studio estiver disponível na máquina.
- e) O Framework .NET só está disponível em sistemas que utilizem o Windows como sistema operacional.

Variáveis

São a estrutura mais básica para armazenar dados.
Alguns dos tipos mais importantes de variáveis são:

Tipo	Natureza	<i>Bits</i>
int	Inteiro	32
long	Inteiro	64
float	Ponto flutuante	32
double	Ponto flutuante	64
char	Caractere	16
string	Caracteres	16 por caractere
bool	Verdadeiro ou falso	8

Nomes para variáveis

- O primeiro caractere deve ser uma letra.
- Se usarmos mais de um caractere, podemos usar letras ou algarismos.
- O C# diferencia letras maiúsculas de minúsculas, por isso, a variável *media* é diferente de *Media*, que também é diferente de *MEDIA*.
- Não utilize caracteres especiais, como acentos (á, é â, ã, ç etc.).
- Não utilize palavras reservadas da linguagem, como nomes de variáveis.
- Recomenda-se o uso de nomes mnemônicos.

Declaração de variáveis

Em C#, variáveis são declaradas escrevendo seu tipo, seguido do nome da variável:

```
int x;  
bool opcao;  
string texto1;
```

Uma vez declaradas, valores podem ser atribuídos às variáveis:

```
x = 10;  
opcao = false;  
texto1 = "Boa noite";
```

Declaração e atribuição de variáveis

- Podemos declarar variáveis e atribuir um valor inicial a elas em uma mesma linha:

```
int x = 10;  
bool opcao = false;  
string texto1 = "Boa noite";
```



Operações aritméticas

Operação	Nome da operação	Exemplo
+	Soma	$7 + 3 = 10$
-	Subtração	$7 - 3 = 4$
*	Multiplicação	$7 * 3 = 21$
/	Divisão	$7 / 3 = 2$
%	Resto da Divisão	$7 \% 3 = 1$

Operadores unários e de atribuição

Operação	Significado	Sinônimo
$a++$	Incremento	$a = a + 1$
$a--$	Decremento	$a = a - 1$
$a += 5$	Atribuição de soma	$a = a + 5$
$a -= 5$	Atribuição de subtração	$a = a - 5$
$a *= 5$	Atribuição de multiplicação	$a = a * 5$
$a /= 5$	Atribuição de divisão	$a = a / 5$
$a \% = 5$	Atribuição de resto	$a = a \% 5$

Operações aritméticas – exemplo

```
int a = 7;  
int b = 3;  
  
int soma = a + b;  
int subtracao = a - b;  
int multiplicacao = a * b;  
int divisao = a / b;  
int resto = a % b;
```

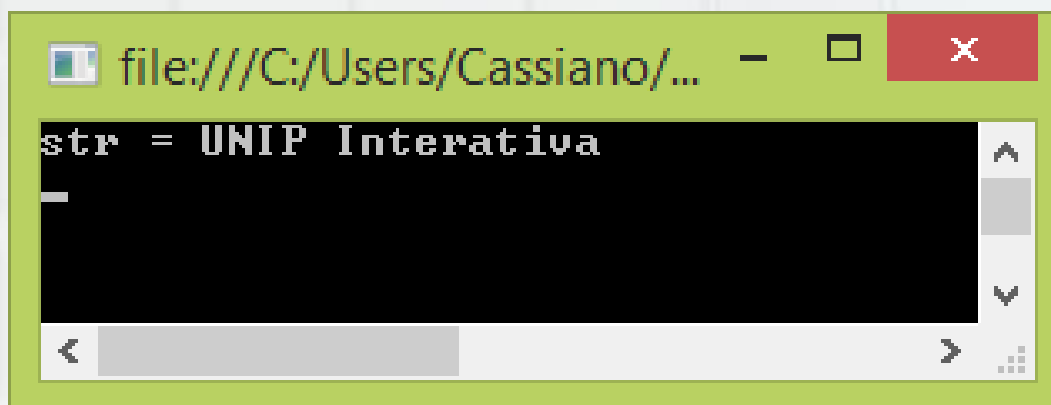


Concatenação de strings

```
void Main (string [] args)
{
    string str1 = "UNIP";
    string str2 = " Interativa";

    string str = str1 + str2;

    Console.WriteLine("str = {0}", str);
}
```



A screenshot of a Windows console window with a green title bar. The title bar text is "file:///C:/Users/Cassiano/...". The console output shows "str = UNIP Interativa" on the first line, followed by a blank line. The window has standard Windows window controls (minimize, maximize, close) and a scrollbar on the right.

Entrada de dados e conversão de tipos

```
11 static void Main(string[] args)
12 {
13     string nome;
14     int idade;
15     Console.WriteLine("Olá, qual o seu nome?");
16     nome = Console.ReadLine();
17
18     Console.WriteLine();
19     Console.WriteLine("E qual é a sua idade?");
20     // a idade é lida como um string e convertida para int
21     idade = Convert.ToInt32(Console.ReadLine());
22
23     Console.WriteLine();
24     Console.WriteLine("{0}, você tem {1} anos de idade.", nome, idade);
25
26     Console.ReadKey();
27 }
```

Métodos para fazer conversão de tipos

Tipo numérico	Método <i>Convert</i>	Método <i>Parse</i>
short	Convert.ToInt16(String)	Int16.Parse(String)
int	Convert.ToInt32(String)	Int32.Parse(String)
long	Convert.ToInt64(String)	Int64.Parse(String)
float	Convert.ToSingle(String)	Single.Parse(String)
double	Convert.ToDouble(String)	Double.Parse(String)

Interatividade

Qual das linhas abaixo faz uma declaração de variável de maneira correta?

- a) `a = int 5;`
- b) `float a = "5";`
- c) `double = 5;`
- d) `short 5a;`
- e) `string a = "5";`



Resposta

Qual das linhas abaixo faz uma declaração de variável de maneira correta?

- a) `a = int 5;`
- b) `float a = "5";`
- c) `double = 5;`
- d) `short 5a;`
- e) `string a = "5";`



Manipulação de strings

Variáveis (e constantes) do tipo string possuem os seguintes membros que ajudam a manipular o string contido nelas:

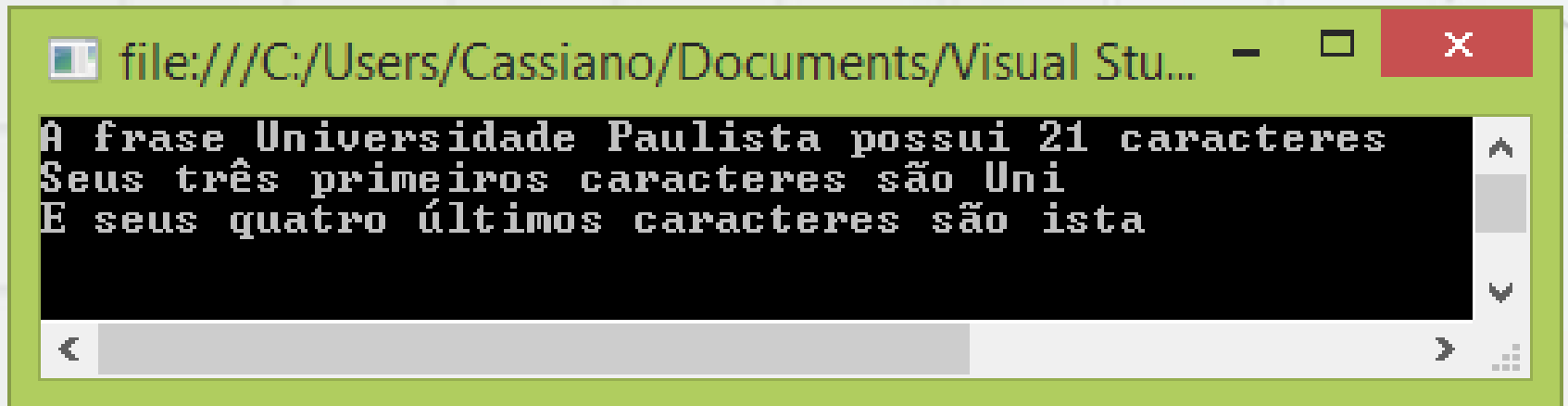
- **Length** – é um atributo que contém o tamanho do string;
- **Substring (int posicaoInicial, int comprimento)** – este método permite obter um substring, ou seja, um pedaço do string original. Seus parâmetros são posicaoInicial, que indica qual o primeiro caractere do substring, e comprimento, que indica qual o tamanho do substring.

Manipulação de strings

```
static void Main(string[] args)
{
    string frase = "Universidade Paulista";

    Console.WriteLine("A frase {0} possui {1} caracteres", frase, frase.Length);
    Console.WriteLine("Seus três primeiros caracteres são {0}", frase.Substring(0, 3));
    Console.WriteLine("E seus quatro últimos caracteres são {0}", frase.Substring(frase.Length - 4, 4));

    Console.ReadKey();
}
```



The screenshot shows a Windows command prompt window with a green title bar. The title bar text is "file:///C:/Users/Cassiano/Documents/Visual Stu...". The window contains the following output from the C# program:

```
A frase Universidade Paulista possui 21 caracteres
Seus três primeiros caracteres são Uni
E seus quatro últimos caracteres são ista
```

The output is displayed in a monospaced font on a black background. The window has standard Windows window controls (minimize, maximize, close) in the title bar and a scrollbar on the right side.

Desvio condicional

```
11 static void Main(string[] args)
12 {
13     double numero = 0;
14     Console.Write("Digite um número: ");
15     numero = Convert.ToDouble(Console.ReadLine());
16
17     if (numero % 2 == 0)
18     {
19         Console.WriteLine("O número {0} é par.", numero);
20     }
21     else
22     {
23         Console.WriteLine("O número {0} é ímpar.", numero);
24     }
25     Console.ReadKey();
26 }
```

Operadores relacionais

Operador	Descrição
==	Igual a
!=	Diferente de
>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a

Operadores lógicos

Operador	Descrição
&&	E
	Ou
!	Não

Exemplo de expressão lógica composta

```
11 static void Main(string[] args)
12 {
13     double numero = 0;
14     Console.Write("Digite um número: ");
15     numero = Convert.ToDouble(Console.ReadLine());
16
17     if (numero % 2 == 0 && numero > 5)
18     {
19         Console.WriteLine("O número {0} é par e é maior que 5.", numero);
20     }
21     else if (numero % 2 == 0 && numero <= 5)
22     {
23         Console.WriteLine("O número {0} é par e é menor ou igual a 5.", numero);
24     }
25     else
26     {
27         Console.WriteLine("O número {0} é ímpar.", numero);
28     }
29     Console.ReadKey();
30 }
```

Desvio condicional tipo switch-case

```
11 static void Main(string[] args)
12 {
13     int numero = 0;
14     Console.Write("Digite um número: ");
15     numero = Convert.ToInt32(Console.ReadLine());
16
17     switch (numero)
18     {
19         case 1:
20             Console.WriteLine("Você digitou um");
21             break;
22         case 2:
23             Console.WriteLine("Você digitou dois");
24             break;
25         case 3:
26             Console.WriteLine("Você digitou três");
27             break;
28         default:
29             Console.WriteLine("Você digitou outro valor");
30             break;
31     }
32     Console.ReadKey();
33 }
```

Interatividade

Qual será a saída gerada por este código?

```
static void Main(string[] args)
{
    int numero = 7;
    bool numeroPar = numero % 2 == 0;
    bool menorQueCinco = numero < 5;

    if (numeroPar || menorQueCinco)
    {
        Console.WriteLine("O número é par e é menor que cinco");
    }
    Console.ReadKey();
}
```

a) Nada.

b) true.

c) false.

d) “O número é par e é menor que cinco”.

e) O programa apresenta erros.

Resposta

Qual será a saída gerada por este código?

```
static void Main(string[] args)
{
    int numero = 7;
    bool numeroPar = numero % 2 == 0;
    bool menorQueCinco = numero < 5;

    if (numeroPar || menorQueCinco)
    {
        Console.WriteLine("O número é par e é menor que cinco");
    }
    Console.ReadKey();
}
```

a) Nada.

b) true.

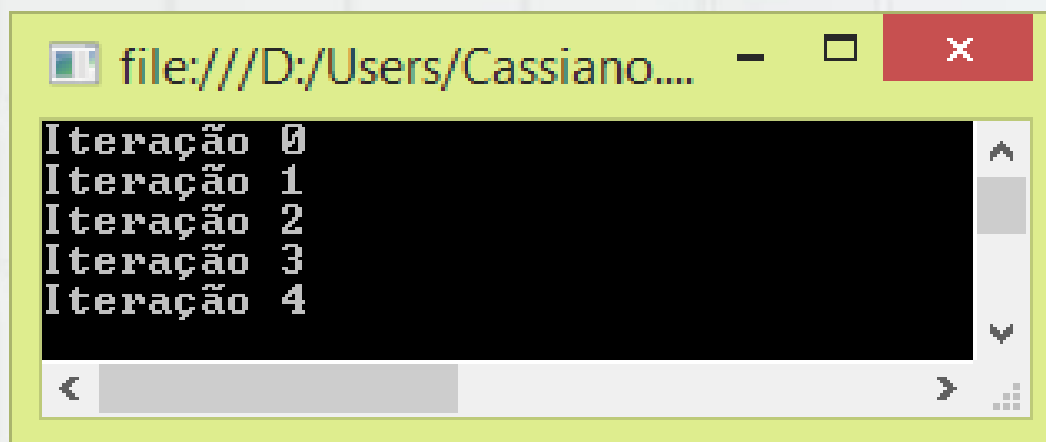
c) false.

d) “O número é par e é menor que cinco”.

e) O programa apresenta erros.

Laço – for

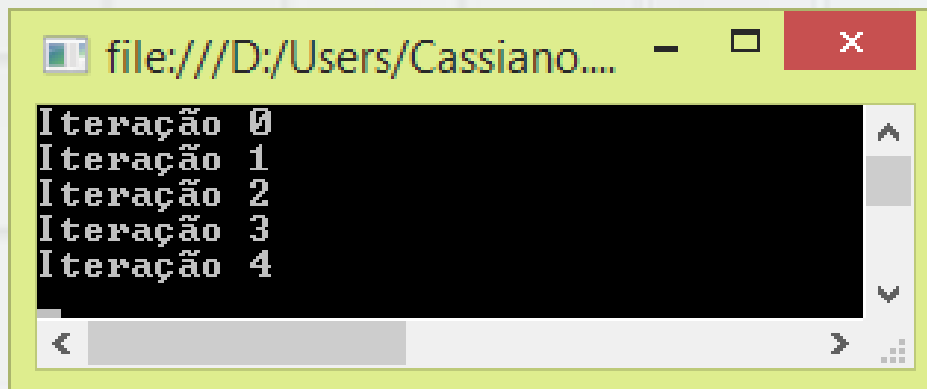
```
static void Main(string[] args)
{
    for (int i = 0; i < 5; i++)
    {
        Console.WriteLine("Iteração {0}", i);
    }
    Console.ReadKey();
}
```



```
file:///D:/Users/Cassiano...
Iteração 0
Iteração 1
Iteração 2
Iteração 3
Iteração 4
```


Laço – while

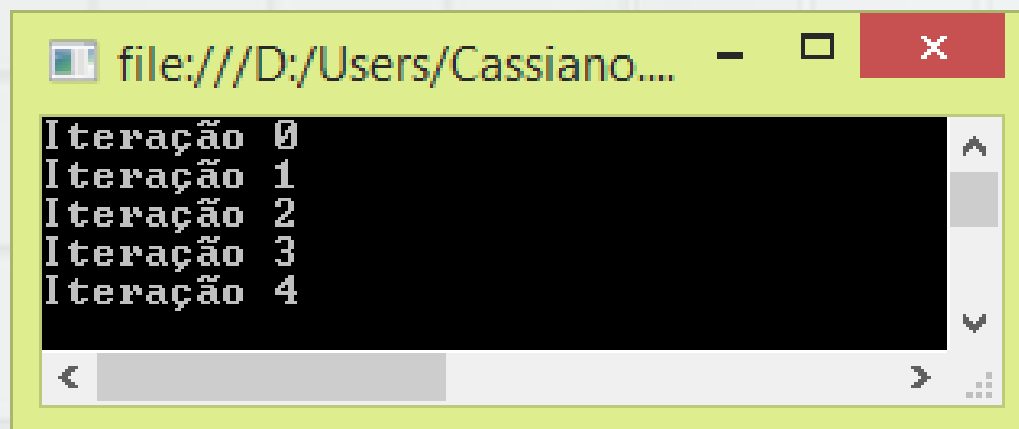
```
static void Main(string[] args)
{
    int i = 0;
    while (i < 5)
    {
        Console.WriteLine("Iteração {0}", i);
        i++;
    }
    Console.ReadKey();
}
```

A screenshot of a Windows console window. The title bar is green and shows the file path "file:///D:/Users/Cassiano....". The console output displays five lines: "Iteração 0", "Iteração 1", "Iteração 2", "Iteração 3", and "Iteração 4". The text is white on a black background. There are scroll bars on the right and bottom of the console area.

```
file:///D:/Users/Cassiano....
Iteração 0
Iteração 1
Iteração 2
Iteração 3
Iteração 4
```

Laço – do-while

```
static void Main(string[] args)
{
    int i = 0;
    do
    {
        Console.WriteLine("Iteração {0}", i);
        i++;
    } while (i < 5);
    Console.ReadKey();
}
```

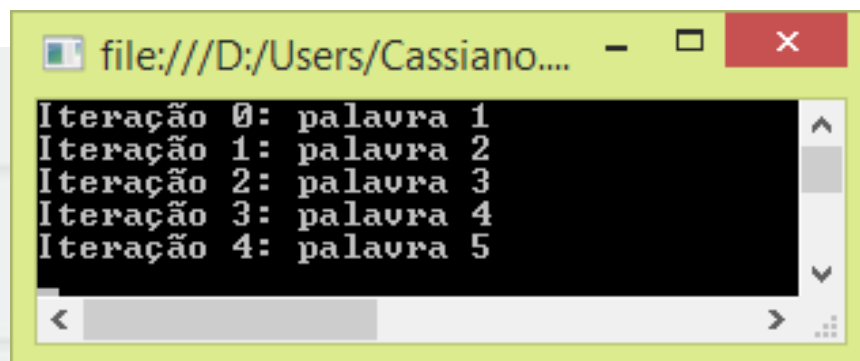


The screenshot shows a Windows console window with a yellow title bar. The title bar text is "file:///D:/Users/Cassiano...". The console output displays five lines: "Iteração 0", "Iteração 1", "Iteração 2", "Iteração 3", and "Iteração 4". The text is in a monospaced font. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

Laço – foreach

```
static void Main(string[] args)
{
    string[] texto = new string[5];
    texto[0] = "palavra 1";
    texto[1] = "palavra 2";
    texto[2] = "palavra 3";
    texto[3] = "palavra 4";
    texto[4] = "palavra 5";
    int contador = 0;

    foreach (string palavra in texto)
    {
        Console.WriteLine("Iteração {0}: {1}", contador, palavra);
        contador++;
    }
    Console.ReadKey();
}
```

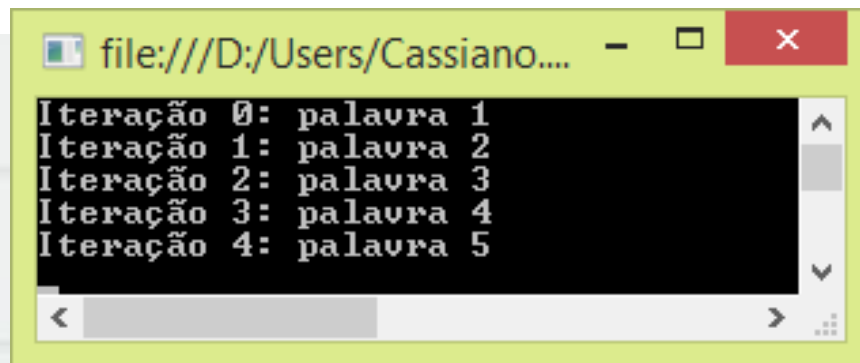


```
file:///D:/Users/Cassiano....
Iteração 0: palavra 1
Iteração 1: palavra 2
Iteração 2: palavra 3
Iteração 3: palavra 4
Iteração 4: palavra 5
```

Laço – foreach usando uma lista dinâmica

```
static void Main(string[] args)
{
    List<string> texto = new List<string>();
    texto.Add("palavra 1");
    texto.Add("palavra 2");
    texto.Add("palavra 3");
    texto.Add("palavra 4");
    texto.Add("palavra 5");
    int contador = 0;

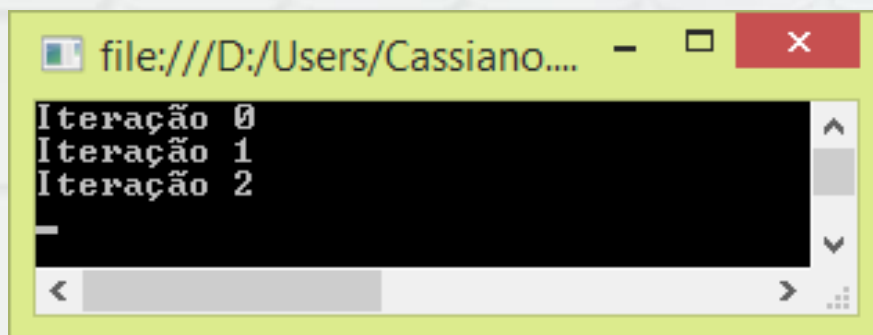
    foreach (string palavra in texto)
    {
        Console.WriteLine("Iteração {0}: {1}", contador, palavra);
        contador++;
    }
    Console.ReadKey();
}
```



```
file:///D:/Users/Cassiano....
Iteração 0: palavra 1
Iteração 1: palavra 2
Iteração 2: palavra 3
Iteração 3: palavra 4
Iteração 4: palavra 5
```

Interrupções de laços – break

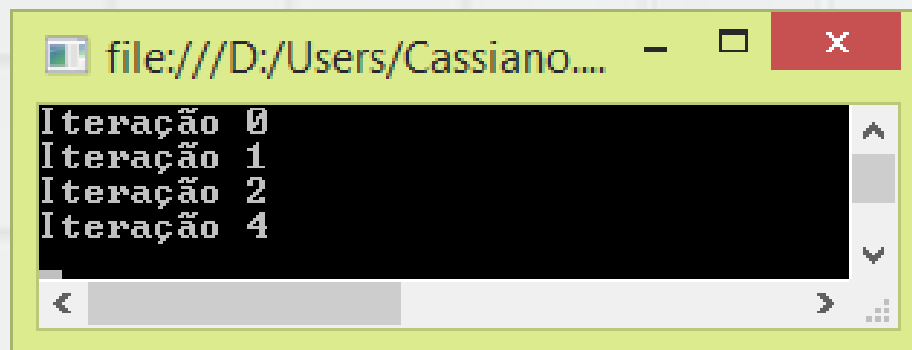
```
static void Main(string[] args)
{
    for (int i = 0; i < 5; i++)
    {
        if (i == 3)
        {
            break;
        }
        Console.WriteLine("Iteração {0}", i);
    }
    Console.ReadKey();
}
```



```
file:///D:/Users/Cassiano...
Iteração 0
Iteração 1
Iteração 2
-
```

Interrupções de laços – continue

```
static void Main(string[] args)
{
    for (int i = 0; i < 5; i++)
    {
        if (i == 3)
        {
            continue;
        }
        Console.WriteLine("Iteração {0}", i);
    }
    Console.ReadKey();
}
```



```
file:///D:/Users/Cassiano....
Iteração 0
Iteração 1
Iteração 2
Iteração 4
```

Interatividade

Qual das seguintes declarações do laço for está incorreta?

- a) for (int i = 5; i > 0; i--)
- b) for (; i > 5; i++)
- c) for (int i = 0; ; i += 5)
- d) for (int i = 0; i > 5;)
- e) for (int i = 0; i +5; i += 5)



Resposta

Qual das seguintes declarações do laço for está incorreta?

- a) for (int i = 5; i > 0; i--)
- b) for (; i > 5; i++)
- c) for (int i = 0; ; i += 5)
- d) for (int i = 0; i > 5;)
- e) for (int i = 0; i +5; i += 5)



ATÉ A PRÓXIMA!



Interativa