



Willian Menezes

Treinamento de C# e .NET

Básico/Intermediário

Namespaces

Separacao logica

Dominio

Classe

Classe

Classe

Classe

Servicos

Classe

Classe

Classe

Classe

Namespaces



**Mesmo nome
namespace diferente**

Isso eh bom?

Namespaces

Nomeando um namespace

```
namespace CursoCsharp.Servicos.Jeta
{
    // ...
}
```

0 references

P **D** **D**

P - Nome do projeto

D - Nome do Diretorio

Por convensao apenas um namespace por aquivo
Podemos ter o mesmo namespace em arquivos diferentes

Using

Estamos falando que queremos usar as funcionalidades dentro do namespace indicado

Namespace -> System

Classe -> Console

Metodo -> WriteLine("texto");

Sem o using -> System.Console.WriteLine("meu texto").

E o global using?

Definicao de namespaces

```
namespace CursoCsharp.Servicos;

0 references
internal class ServicoTeste
{
    // ...
}

namespace CursoCsharp.Servicos3;
0 references
internal class Servico2Teste
{
    // ...
}
```

dotnet 6

Não precisa fechar chaves

Apenas um por arquivo

Nome: file-scoped namespaces

Mas Isso eh bom?

Assembly

Uma DLL ou EXE | Um agrupamento físico das classes relacionadas

Pagamento

Dominio

Classe

Classe

Classe

Classe

Servicos

Classe

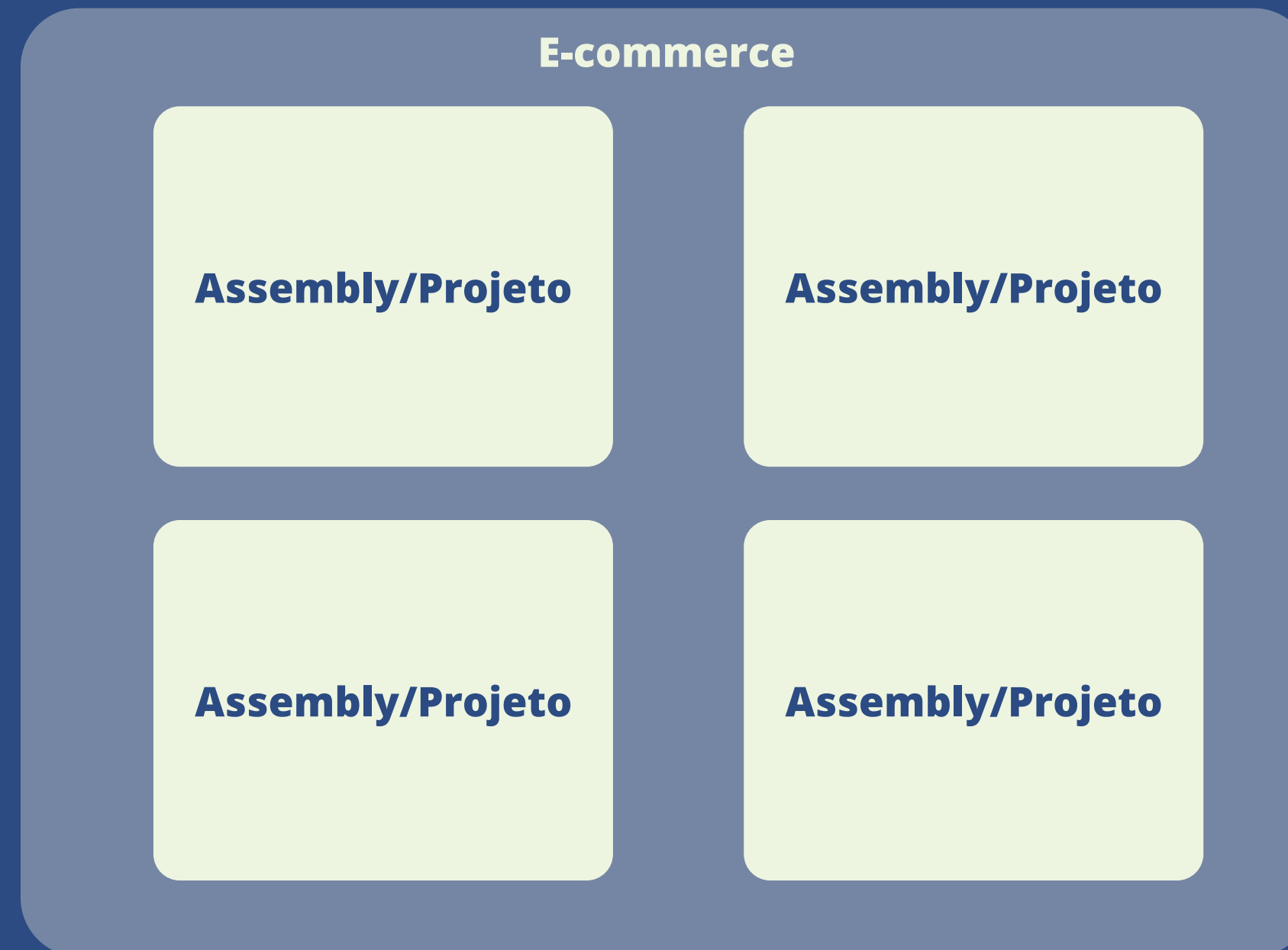
Classe

Classe

Classe

Aplicacao

Varios assemblies ou projetos



Solution = Aplicacao
Project = Assembly

Criando nosso primeiro programa

Convenções

- Não é permitido começar com dígitos
- Não pode utilizar acentos e til
- Não pode ter espaço em branco
- Camel Case
 - Parâmetros de métodos
 - variáveis no corpo de métodos
- Pascal Case
 - Classes
 - Metodos
 - Propriedades

Palavras reservadas

Podemos escrever tudo... respeitando algumas regras

Case sensitive

Palavras reservadas (keywords)

Podemos usar keywords se utilizarmos o @ (Não façam isso em casa)

Temos a ajuda da IDE

```
0 references
static void Main(string[] args)
{
    Console.WriteLine("Texto");

    Console.writeLine("Texto");
}
```

```
0 references
static void Main(string[] args)
{
    int numero = 10;
    int @int = 10;

    int int = 10;
    int using = 10;
    int string = 10;
}
```

Tipos primitivos

Características dos tipos integrais

O C# é compatível com os seguintes tipos integrais predefinidos:

palavra-chave/tipo C#	Intervalo	Tamanho	Tipo .NET
sbyte	-128 a 127	Inteiro de 8 bits com sinal	System.SByte
byte	0 a 255	Inteiro de 8 bits sem sinal	System.Byte
short	-32.768 a 32.767	Inteiro de 16 bits com sinal	System.Int16
ushort	0 a 65.535	Inteiro de 16 bits sem sinal	System.UInt16
int	-2.147.483.648 a 2.147.483.647	Inteiro assinado de 32 bits	System.Int32
uint	0 a 4.294.967.295	Inteiro de 32 bits sem sinal	System.UInt32
long	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807	Inteiro assinado de 64 bits	System.Int64
ulong	0 a 18.446.744.073.709.551.615	Inteiro de 64 bits sem sinal	System.UInt64
nint	Depende da plataforma (computada em runtime)	Inteiro de 32 bits ou de 64 bits com sinal	System.IntPtr
nuint	Depende da plataforma (computada em runtime)	Inteiro de 32 bits ou de 64 bits sem sinal	System.UIntPtr

A seguinte tabela mostra os valores padrão de tipos C#:

Type	Valor padrão
Qualquer tipo de referência	null
Qualquer tipo numérico integral interno	0 (zero)
Qualquer tipo numérico de ponto flutuante interno	0 (zero)
bool	false
char	'\0' (U+0000)
enumeração	O valor é produzido pela expressão (E)0, em que E é o identificador de enumeração.
struct	O valor produzido pela configuração de todos os campos tipo-valor para seus valores padrão e todos os campos tipo-referência para null.
Qualquer tipo de valor que permite valor nulo	Uma instância para a qual a propriedade HasValue é false e a propriedade Value não está definida. Esse valor padrão também é conhecido como o valor null do tipo de valor que permite valor nulo.

Características dos tipos de ponto flutuante

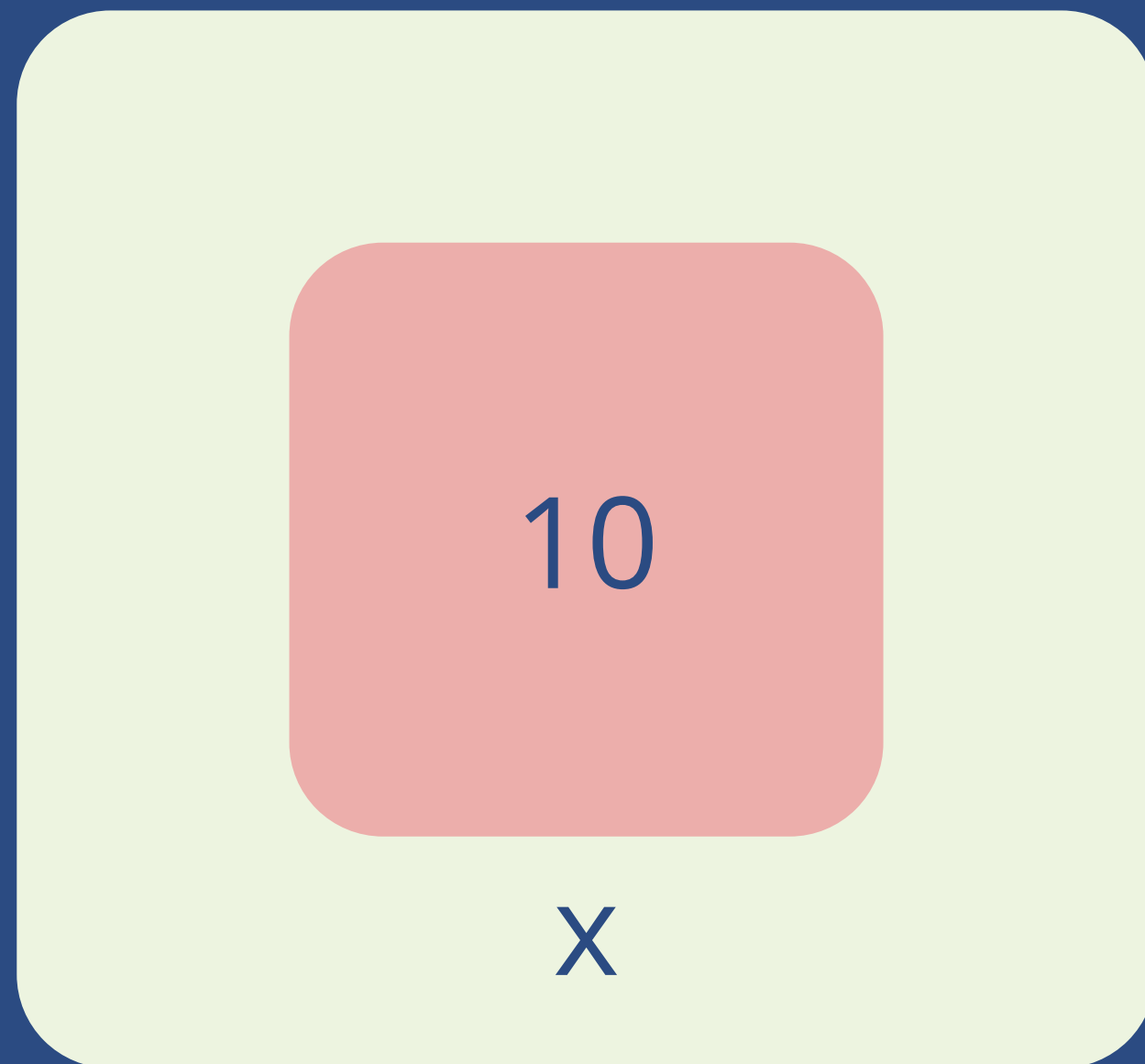
O C# é compatível com os seguintes tipos de pontos flutuantes predefinidos:

palavra-chave/tipo C#	Intervalo aproximado	Precisão	Tamanho	Tipo .NET
float	±1,5 x 10 ⁻⁴⁵ para ±3,4 x 10 ³⁸	~6 a 9 dígitos	4 bytes	System.Single
double	±5.0 × 10 ⁻³²⁴ to ±1.7 × 10 ³⁰⁸	~15 a 17 dígitos	8 bytes	System.Double
decimal	±1,0 x 10 ⁻²⁸ para ±7,9228 x 10 ²⁸	28 a 29 dígitos	16 bytes	System.Decimal

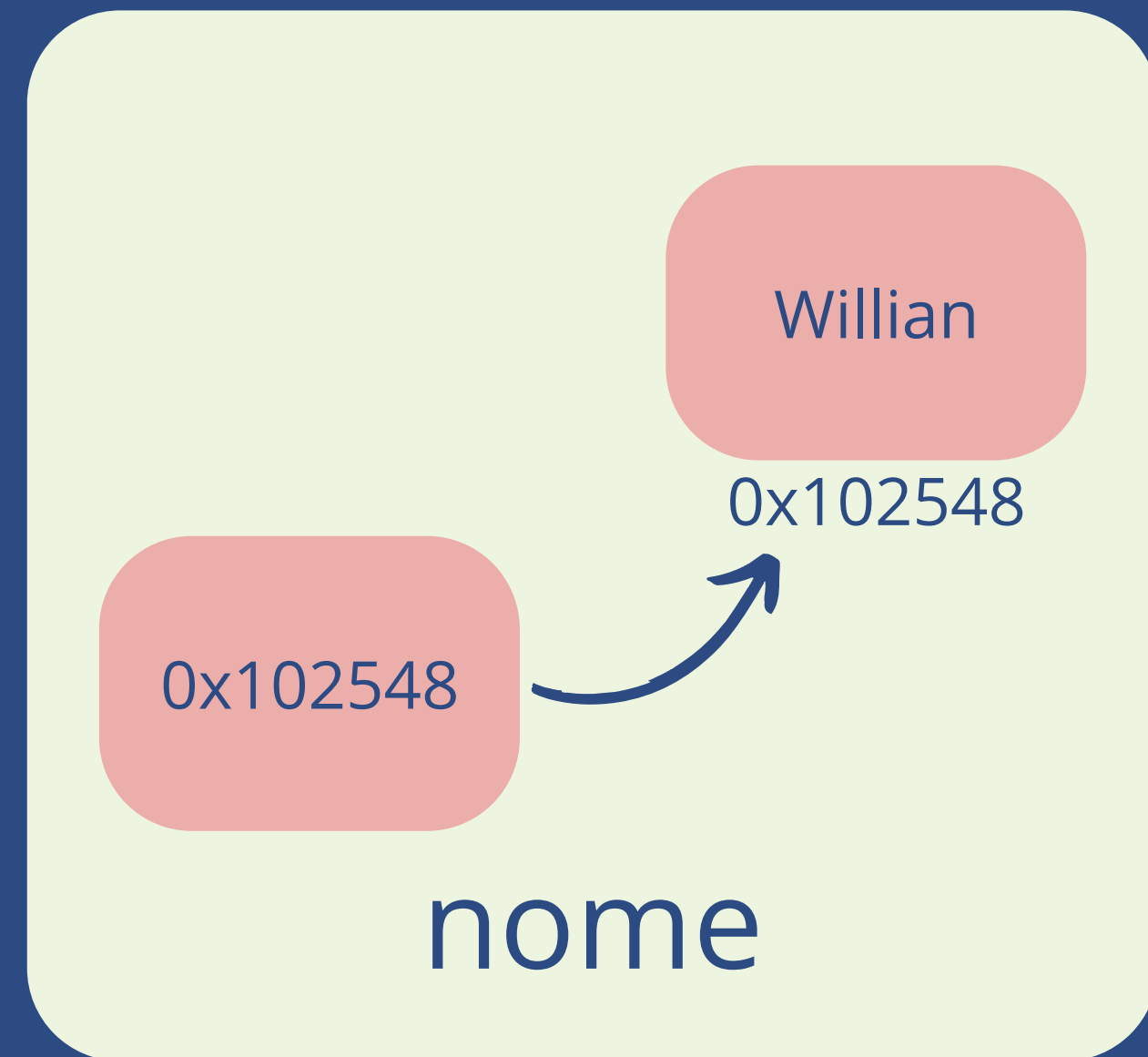
Tipo	Intervalo	Tamanho	Tipo .NET
char	U+0000 a U+FFFF	16 bits	System.Char

Tipos valor x Referencia

```
int numero = 20;
```



```
string nome = "Willian";
```



Tipos de valor

- Contem diretamente a instancia do seu tipo
- Nao alocam na heap e nao acionam o coletor de lixo
- Na atribuicao de outra variavel seu valor eh copiado
- Nao podem receber null

Existem duas categorias

Struct

- Propriedades e métodos
- Não podemos realizar herança
- Podemos criar nossos próprios tipos de valor

Enum

- Conjunto de constantes integrais nomeadas
- Muito mais intuitivo em alguns casos
- Herdam de System.Enum que herda de System.ValueType

Como colocar os valores em tela?

Demo

```
Console.WriteLine();  
Console.Write();
```

Formatando numeros e casas
decimais

.ToString("F1")
CultureInfo?

Concatenando strings

Interpolacao, placeholder e
concatenacao

Operadores de atribuição

Atribui o valor do lado direito da variavel:

int x = 10;

x recebe o valor 10;

```
// estou atribuindo o valor 10 a variavel x.  
int x = 10;  
  
// estou atribuindo um conjunto de caracteres a variavel nome.  
string nome = "Willian Menezes";
```

Atribuicao composta

Operador aritmetico + Operador de atribuicao

```
int a = 5;  
a += 9;  
Console.WriteLine(a); // output: 14  
  
a -= 4;  
Console.WriteLine(a); // output: 10  
  
a *= 2;  
Console.WriteLine(a); // output: 20  
  
a /= 4;  
Console.WriteLine(a); // output: 5  
  
a %= 3;  
Console.WriteLine(a); // output: 2
```

Exercicio 01

Escreva um programa que produza a saída:

O Willian possui 26 anos.

O Matheus possui 22 anos.

Escreva um programa que produza a saída:

O valor de um PCGamer e R\$ 3000,00;

O valor de uma cadeira de escritório e R\$ 1700,00;

Conversões

Implicita

Não precisamos informar como sera feita
Tem que ser uma conversão possível
O compilador faz isso por nos

```
int valor = 1050;

// convertendo int para double
double outroValor = valor;

Console.WriteLine(valor);
Console.WriteLine(outroValor);
```

```
double valor = 10.50;

// erro - nao podemos converter implicitamente double para int
int outroValor = valor;

Console.WriteLine(valor);
Console.WriteLine(outroValor);
```

Explicita

Precisamos explicitar a conversão que queremos fazer (casting)

```
double valor = 10.50;

// valor convertido, porem perdeu sua precisao
int outroValor = (int)valor;

Console.WriteLine(valor);
Console.WriteLine(outroValor);
```

Outras conversões

```
int numero = int.Parse("10");
int outroNumero = Convert.ToInt32("10");
int outroNumeroQualquer = Convert.ToInt32(10.2);
```


Operadores Aritméticos

Operador de incremento/decremento unario

```
// Pre-fixado
double a = 1.5;
Console.WriteLine(a);    // output: 1.5
Console.WriteLine(++a);  // output: 2.5
Console.WriteLine(a);    // output: 2.5

// Pos-fixado
int i = 3;
Console.WriteLine(i);     // output: 3
Console.WriteLine(i++);   // output: 3
Console.WriteLine(i);     // output: 4
```

Operadores Aritméticos

OBS

Multiplicação, divisão e mod tem precedência maior que adição e subtração

Adição +
Subtração -
Multiplicação *
Divisão /
Mod(Resto da divisão) %

```
Console.WriteLine(5 % 4);    // output: 1  
Console.WriteLine(5 % 3);    // output: 2
```

Tambem podemos concatenar string

```
Console.WriteLine(5 + 4);      // output: 9  
Console.WriteLine(5 + 4.3);    // output: 9.3  
Console.WriteLine(5.1m + 4.2m); // output: 9.3
```

```
Console.WriteLine(47 - 3);      // output: 44  
Console.WriteLine(5 - 4.3);     // output: 0.7  
Console.WriteLine(7.5m - 2.3m); // output: 5.2
```

```
Console.WriteLine(5 * 2);        // output: 10  
Console.WriteLine(0.5 * 2.5);    // output: 1.25  
Console.WriteLine(0.1m * 23.4m); // output: 2.34
```

```
Console.WriteLine(13 / 5);       // output: 2  
Console.WriteLine(-13 / 5);      // output: -2  
Console.WriteLine(13 / -5);      // output: -2  
Console.WriteLine(-13 / -5);     // output: 2
```

```
Console.WriteLine(13 / 5.0);     // output: 2.6
```

```
int a = 13;  
int b = 5;  
Console.WriteLine((double)a / b); // output: 2.6
```

Entrada de dados

`Console.ReadLine();`

Lê os dados e retorna em formato string

Ler e mostra o nome de uma pessoa

Ler dois números inteiros e mostrar sua soma

Ler dois números de ponto flutuante e mostrar sua soma

Ler o nome, idade, sexo e altura(em cm) na mesma linha e mostrar os dados.

Exercicio 02

Faça um Programa que peça as 4 notas bimestrais e mostre a média.

Faça um Programa que pergunte quanto você ganha por hora e o número de horas trabalhadas no mês. Calcule e mostre o total do seu salário no referido mês.

Faça um Programa que pergunte quanto você ganha por hora e o número de horas trabalhadas no mês. Calcule e mostre o total do seu salário no referido mês, sabendo-se que são descontados 11% para o Imposto de Renda, 8% para o INSS e 5% para o sindicato, faça um programa que nos dê:

- a. Salário bruto.
- b. Quanto pagou ao INSS.
- c. Quanto pagou ao sindicato.
- d. o salário líquido.
- e. calcule os descontos e o salário líquido, conforme a tabela abaixo

Saida:

+ Salário Bruto : R\$

- IR (11%) : R\$

- INSS (8%) : R\$

- Sindicato (5%) : R\$

= Salário Liquido : R\$

Operadores de comparação

Igualdade e desigualdade (Tipos de valor)

```
int a = 1 + 2 + 3;
int b = 6;
Console.WriteLine(a == b); // output: True

int a = 1 + 2 + 3;
int b = 6;
Console.WriteLine(a != b); // output: False

char c1 = 'a';
char c2 = 'A';
Console.WriteLine(c1 == c2); // output: False
Console.WriteLine(c1 == char.ToLower(c2)); // output: True
```

```
Console.WriteLine(7.0 < 5.1); // output: False
Console.WriteLine(5.1 < 5.1); // output: False
Console.WriteLine(0.0 < 5.1); // output: True
```

```
Console.WriteLine(double.NaN < 5.1); // output: False
Console.WriteLine(double.NaN >= 5.1); // output: False
```

```
Console.WriteLine(7.0 > 5.1); // output: True
Console.WriteLine(5.1 > 5.1); // output: False
Console.WriteLine(0.0 > 5.1); // output: False
```

```
Console.WriteLine(double.NaN > 5.1); // output: False
Console.WriteLine(double.NaN <= 5.1); // output: False
```

```
Console.WriteLine(7.0 <= 5.1); // output: False
Console.WriteLine(5.1 <= 5.1); // output: True
Console.WriteLine(0.0 <= 5.1); // output: True
```

```
Console.WriteLine(double.NaN > 5.1); // output: False
Console.WriteLine(double.NaN <= 5.1); // output: False
```

```
Console.WriteLine(7.0 >= 5.1); // output: True
Console.WriteLine(5.1 >= 5.1); // output: True
Console.WriteLine(0.0 >= 5.1); // output: False
```

```
Console.WriteLine(double.NaN < 5.1); // output: False
Console.WriteLine(double.NaN >= 5.1); // output: False
```


Operadores lógicos

Operador and (&&)

```
int x = 10;
int y = 20;

bool resultado = x >= 10 && y <= 20; // True

bool outroResultado = x < 10 && y <= 20; // False
```

No segundo exemplo o operando a direita não é avaliado, porque o primeiro operando retornou false, logo o resultado no "&&" será false.

Operador OU (||)

```
int x = 10;
int y = 20;

bool resultado = x >= 10 || y > 20; // True

bool outroResultado = x < 10 || y > 20; // False
```

No primeiro exemplo o operando a direita não é avaliado, porque o primeiro operando retornou true, logo o resultado no "||" será true.

Precedencia: ! > && > ||

Operador de negação (!)

```
bool passed = false;
Console.WriteLine(!passed); // output: True
Console.WriteLine(!true); // output: False
```

Tabela verdade E

C1	C2	&&
F	F	F
F	V	F
V	F	F
V	V	V

Tabela verdade E

C1	C2	
V	V	V
V	F	V
F	V	V
F	F	F

Estrutura condicional

O bloco IF avalia uma expressão booleana e se for verdadeira executa um bloco de comandos.

```
// Utilizando o IF sem o ELSE

var salario = 1000;

if (salario < 500)
{
    Console.WriteLine("Preciso calcular uma regra de imposto.");
}

// continua.....
```

```
var salario = 1000;

string retorno = salario > 500
    ? "Preciso calcular uma regra de imposto."
    : "Nenhuma regra precisa ser aplicada.";

Console.WriteLine(retorno); // saida -> Preciso calcular uma regra de imposto.
```

```
// Utilizando o IF com uma parte ELSE

var salario = 1000;

if (salario > 500)
{
    Console.WriteLine("Preciso calcular uma regra de imposto.");
} else
{
    Console.WriteLine("Nenhuma regra precisa ser aplicada.");
}

// continua.....
```

```
// Utilizando o IF's encadeados

var salario = 1000;

if (salario <= 1000)
{
    Console.WriteLine("Estou ganhando mal.");
} else if (salario > 1000 && salario <= 2000){
    Console.WriteLine("Estou ganhando mais ou menos.");
} else {
    Console.WriteLine("Estou ganhando bem.");
}

// continua.....
```

Estrutura condicional

Podemos criar o switch em C# de várias formas, com o padrão relacional, padrão constante e etc. Mas vamos focar apenas no convencional.

```
var status = 1;

switch (status)
{
    case 1:
        Console.WriteLine("Separar pedido");
        break;
    case 2:
        Console.WriteLine("Embalar pedido");
        break;
    case 3:
        Console.WriteLine("Enviar pedido");
        break;
    default:
        Console.WriteLine("Pedido encerrado");
        break;
}
```


Observacoes:

Não podemos utilizar uma variável que não foi iniciada

Escopo de uma variavel

Estrutura condicional

Exercicios

As Organizações Tabajara resolveram dar um aumento de salário aos seus colaboradores e lhe contrataram para desenvolver o programa que calculará os reajustes.

Faça um programa que recebe o salário de um colaborador e o reajuste segundo o seguinte critério, baseado no salário atual:

- salários até R\$ 280,00 (incluindo) : aumento de 20%
- salários entre R\$ 280,00 e R\$ 700,00 : aumento de 15%
- salários entre R\$ 700,00 e R\$ 1500,00 : aumento de 10%
- salários de R\$ 1500,00 em diante : aumento de 5% Após o aumento ser realizado, informe na tela:
- o salário antes do reajuste;
- o percentual de aumento aplicado;
- o valor do aumento;
- o novo salário, após o aumento.

Faça um Programa que leia um número e exiba o dia correspondente da semana. (1-Domingo, 2- Segunda, etc.), se digitar outro valor deve aparecer valor inválido.

Faça um programa para o cálculo de uma folha de pagamento, sabendo que os descontos são do Imposto de Renda, que depende do salário bruto (conforme tabela abaixo) e 3% para o Sindicato e que o FGTS corresponde a 11% do Salário Bruto, mas não é descontado (é a empresa que deposita). O Salário Líquido corresponde ao Salário Bruto menos os descontos. O programa deverá pedir ao usuário o valor da sua hora e a quantidade de horas trabalhadas no mês.Desconto do IR:

- Salário Bruto até 900 (inclusive) - isento
- Salário Bruto até 1500 (inclusive) - desconto de 5%
- Salário Bruto até 2500 (inclusive) - desconto de 10%
- Salário Bruto acima de 2500 - desconto de 20% Imprima na tela as informações, dispostas conforme o exemplo abaixo. No exemplo o valor da hora é 400 e a quantidade de hora é 10.

Saida:

Salário Bruto: (10 * 400)	: R\$ 4000,00
(-) IR (20%)	: R\$ 800,00
(-) SINDICATO (3%)	: R\$ 120,00
FGTS (11%)	: R\$ 440,00
Total de descontos	: R\$ 920,00
Salário Liquido	: R\$ 3080,00

Funções

Bloco de código que realizar um conjunto de instruções.

Escrever código reutilizável e que faça sentido

Vantagens: Reaproveitamento e modularização

```
public static double AcrescentarAoValor(double valor, int porcentagem)
{
    return valor + (valor * porcentagem / 100);
}
```

```
double valorSalario = 1000;
double resultado = AcrescentarAoValor(valorSalario, 10);

Console.WriteLine($"O valor do salario eh: {resultado}");

// saida -> 1100
```

Funções

Exercícios

Data com mês por extenso. Construa uma função que receba uma data no formato DD/MM/AAAA e devolva uma string no formato DD de mesPorExtenso de AAAA.

Faça um programa com uma função chamada somImposto. A função possui dois parâmetros formais: taxaImposto, que é a quantia de imposto sobre vendas expressa em porcentagem e custo, que é o custo de um item antes do imposto. A função “altera” o valor de custo para incluir o imposto sobre vendas.

Faça um programa que use a função valorPagamento para determinar o valor a ser pago por uma prestação de uma conta. O programa deverá solicitar ao usuário o valor da prestação e o número de dias em atraso e passar estes valores para a função valorPagamento, que calculará o valor a ser pago e devolverá este valor ao programa que a chamou. O programa deverá então exibir o valor a ser pago na tela. O cálculo do valor a ser pago é feito da seguinte forma. Para pagamentos sem atraso, cobrar o valor da prestação. Quando houver atraso, cobrar 3% de multa, mais 0,1% de juros por dia de atraso.

Debug

Atalhos:

F5 - Iniciar ou continuar o debug

F9 - Marcar ou desmarcar breakpoint

F10 - Executar fora da função

F11 - Executar dentro da função

SHIFT + F11 - Sair do método em execução

SHIFT + F5 - Parar o debug

Laços de repetição

```
for (int i = 0; i < 3; i++)  
{  
    Console.WriteLine(i);  
}
```

```
var numeros = new List<int> { 1, 2, 3, 4 };  
foreach (int n in numeros)  
{  
    Console.Write($"{n} ");  
}  
// Output:  
// 1 2 3 4
```

```
int n = 0;  
while (n < 5)  
{  
    Console.Write(n);  
    n++;  
}  
// Output:  
// 01234
```

```
int n = 0;  
do  
{  
    Console.Write(n);  
    n++;  
} while (n < 5);  
// Output:  
// 01234
```

Laços de repetição

Exercícios

Faça um programa para imprimir

```
1
2 2
3 3 3
....
n n n n n n ... n
```

para um n informado pelo usuário. Use uma função que receba um valor n inteiro e imprima até a n-ésima linha. (OLHAR NO COD EXERCICIO EXTRA)

Desenvolva um gerador de tabuada, capaz de gerar a tabuada de qualquer número inteiro entre 1 a 10. O usuário deve informar de qual numero ele deseja ver a tabuada. A saída deve ser conforme o exemplo abaixo: (OLHAR NO COD EXERCICIO EXTRA)

- Tabuada de 5:
- 5 X 1 = 5
- 5 X 2 = 10
- ...
- 5 X 10 = 50

O cardápio de uma lanchonete é o seguinte:

Especificação	Código	Preço
Cachorro Quente	100	R\$ 1,20
Bauru Simples	101	R\$ 1,30
Bauru com ovo	102	R\$ 1,50
Hambúrguer	103	R\$ 1,20
Cheeseburger	104	R\$ 1,30
Refrigerante	105	R\$ 1,00

Faça um programa que leia o código dos itens pedidos e as quantidades desejadas. Calcule e mostre o valor a ser pago por item (preço * quantidade) e o total geral do pedido. Considere que o cliente deve informar quando o pedido deve ser encerrado.

Duvidas???