TIPOS DE DADOS (pág.97)

Tipos Inteiros

Tipo	Faixa	Bytes
sbyte	-128 to 127	Com sinal 8-bit
byte	0 to 255	Sem sinal 8-bit
char	U+0000 to U+ffff	Unicode 16-bit character
short	-32,768 to 32,767	Com sinal 16-bit
ushort	0 to 65,535	Sem sinal 16-bit
int	-2,147,483,648 to 2,147,483,647	Com sinal 32-bit
uint	0 to 4,294,967,295	Sem sinal 32-bit
long	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	Com sinal 64-bit
ulong	0 to 18,446,744,073,709,551,615	Sem sinal 64-bit

Tipos com Ponto Flutuante

Tipo	Faixa	Precisão
float	±1.5e−45 to ±3.4e38	7 digits
double	±5.0e−324 to ±1.7e308	15-16 digits

Tipo	Faixa	Precisão	.NET Framework
decimal	±1.0 × 10e−28 to ±7.9 × 10e28	28-29 significant digits	System.Decimal

Outros

Tipo	Descrição	Forma Acesso
string	Texto Unicode	Por referência
bool	true ou false	Por valor
object	Qualquer tipo de objeto do .Net (semelhante a VARIANT) de outras linguagens	Por referência
class	Classe de objeto	referência

DECLARAÇÃO DE VARIÁVEL (pág. 99)

```
int idade;
string nome = "CARLOS MAGNO";
float salario;
float preco = 25.50;
bool casado = false;
char sexo = 'F';
```

CONVENÇÕES PARA NOMES DE VARIÁVEIS (pág. 98)

REGRAS

- Os nomes das variáveis e qualquer outro identificador é "case-sensitive"
- O nome de uma variável deve começar por uma letra, seguida de letras, números ou sublinhados.

CONVENÇÕES

- Apesar de ser permitido, evite utilizar sublinhado no nome de uma variável.
- O nome de uma variável deve começar por uma letra minúscula.
- Se o nome de uma variável for composto de mais de uma palavra coloque a primeira letra de cada palavra (exceto a primeira letra do nome da variável) em maiúsculo e o restante em minúsculo.

  Exemplos:

  nomeDoAluno, precoVenda, dataNascimento, salarioLiquido

OPERADORES

Atribuição (pág. 105)

Operador	Descrição	Preced.
=	Atribuição simples	25
+=	Atribuição por adição	26
-=	Atribuição por subtração	26
*=	Atribuição por multiplicação	27
/=	Atribuição por divisão	27

```
int a = 3, b = 5, c = 10, d = 12, e = 15;
a += 5;
b -= 2;
c *= 3;
d /= 4;
e %= 4;
Console.WriteLine("a = {0}, b = {1}, c = {2}, d = {3}, e = {4}", a, b, c, d, e);
Console.ReadLine();
```

Aritméticos (pág. 102 e 106)

Operador	Descrição	Tipo	Preced.
++	Pós-incremento	Unário	1
--	Pós-decremento	Unário	1
++	Pré-incremento	Unário	2
--	Pré-decremento	Unário	2
*****	Multiplicação	2 operandos	13
/	Divisão	2 operandos	13
+	Adição	2 operandos	14
%	Módulo (resto da divisão)	2 operandos	14
-	Subtração	2 operandos	14

Exemplo 1:

```
int a = 3, b = 5, c = 10;
int d = ++a * b-- - c++;
Console.WriteLine("a = {0}, b = {1}, c = {2}, d = {3}", a, b, c, d);
```

Exemplo 2:

```
int a = 3, b = 5, c = 10;
int d = c % a + b * a - c++ - ++a;
int e = c / b * 2;
Console.WriteLine("a = {0}, b = {1}, c = {2}, d = {3}, e = {4}",
                  a, b, c, d, e);
```

Relacionais (pág. 108 e 109)

Operador	Descrição	Tipo	Preced.
<	Menor que	2 operandos	16
>	Maior que	2 operandos	16
>=	Maior ou igual	2 operandos	16
<=	Menor ou igual	2 operandos	16
==	Igual	2 operandos	17
!=	Diferente	2 operandos	17
is	Testa se um objeto é de uma determinada classe	2 operandos	17

Exemplo 1:

```
int a = 3, b = 5, c = 10;
bool d = a > b*2;
bool e = a * c <= b++ * --c;
bool f = a * b == --a * b;
Console.WriteLine("a = {0}, b = {1}, c = {2}, d = {3}, e = {4} f = {5}",
                  a, b, c, d, e, f);
```

Exemplo 2:

```
int a = 3, b = 5, c = 10;
bool d = c > b / 2;
bool e = a + c-- >= --b * c++;
bool f = a * ++b < --a * c;
Console.WriteLine("a = {0}, b = {1}, c = {2}, d = {3}, e = {4} f = {5}",
                  a, b, c, d, e, f);
```

Lógicos (pág. 110)

Operador	Descrição	Tipo	Preced.
!	NÃO lógico (NOT)	2 operandos	8
&&	E lógico (AND)	2 operandos	22
||	OU lógico	2 operandos	23

Exemplo:

```
int a = 3, b = 5, c = 10;
bool d = a > b * 2 && b > c || 2*a <= b;
bool e = a * 2 <= c && c <= 10 && c < 2 * b | true;
bool f = 2 * b == c || !(a + 2 != b) && b > a;
Console.WriteLine("a = {0}, b = {1}, c = {2}, d = {3}, e = {4} f = {5}",
                   a, b, c, d, e, f);
```

Binários

Operador	Descrição	Tipo	Preced.
~	Complemento binário (NOT)	2 operandos	8
>>	Deslocamento de bits para a direita	2 operandos	15
<<	Deslocamento de bits para a esquerda	2 operandos	15
&	E binário (AND)	2 operandos	19
^	OU EXCLUSIVO binário (XOR)	2 operandos	20
|	OU binário (OR)	2 operandos	21

Exemplos:

	128	**64**	**32**	**16**	**8**	**4**	**2**	**1**
a=0xff	1	1	1	1	1	1	1	1
b=0x81	1	0	0	0	0	0	0	1
c=0xf0	1	1	1	1	0	0	0	0
d=0x00	0	0	0	0	0	0	0	0
e=0x03	0	0	0	0	0	0	1	1

```
int a = 0xff, b = 0x81, c = 0xf0, d = 0x00, e = 0x03;
Console.WriteLine("a = 0xff, b = 0x81, c = 0xf0, d = 0x00");
Console.WriteLine();
Console.WriteLine(" a & c = 0x{0}", (a & c).ToString("x"));
Console.WriteLine(" a & d | b = 0x{0}", (a & d | b).ToString("x"));
Console.WriteLine(" a & c | b = 0x{0}", (a & c | b).ToString("x"));
Console.WriteLine(" c | b & a = 0x{0}", (c | b & a).ToString("x"));
Console.WriteLine();
Console.WriteLine(" ~a = 0x{0}", (~a).ToString("x"));
Console.WriteLine(" ~c = 0x{0}", (~c).ToString("x"));
Console.WriteLine(" ~d = 0x{0}", (~d).ToString("x"));
Console.WriteLine();
Console.WriteLine(" e << 1 = 0x{0} ou {1}", (e << 1).ToString("x"),
                  (e << 1).ToString());
Console.WriteLine(" e << 2 = 0x{0} ou {1}", (e << 2).ToString("x"),
                  (e << 2).ToString());
Console.WriteLine(" e << 3 = 0x{0} ou {1}", (e << 3).ToString("x"),
                  (e << 3).ToString());
```

Ternário

Condição ? vlrVerdadeiro : vlrFalso

```
int hora = 13;
Console.WriteLine(hora > 12 ? "Boa Tarde" : "Bom Dia");
Console.ReadLine();
```