1. Tipos de Dados Primitivos (Básicos)

Esses são os tipos mais simples, usados para armazenar valores individuais.

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo | Tamanho | Intervalo de valores | Exemplo |
| bool | 1 byte | true ou false | bool ativo = true; |
| char | 2 bytes | Qualquer caractere Unicode | char letra = 'A'; |
| byte | 1 byte | 0 a 255 | byte idade = 25; |
| sbyte | 1 byte | -128 a 127 | sbyte temperatura = -10; |
| short | 2 bytes | -32.768 a 32.767 | short numero = 1000; |
| ushort | 2 bytes | 0 a 65.535 | ushort porta = 8080; |
| int | 4 bytes | -2.147.483.648 a 2.147.483.647 | int saldo = 5000; |
| uint | 4 bytes | 0 a 4.294.967.295 | uint positivo = 123456; |
| long | 8 bytes | -9 quintilhões a 9 quintilhões | long populacao = 7800000000; |
| ulong | 8 bytes | 0 a 18 quintilhões | ulong dist = 10000000000; |
| float | 4 bytes | ~7 dígitos de precisão | float pi = 3.14f; |
| double | 8 bytes | ~15-16 dígitos de precisão | double preco = 10.99; |
| decimal | 16 bytes | ~28-29 dígitos de precisão (usado para dinheiro) | decimal valor = 199.99m; |

2. Variáveis e Constantes

* Variáveis: armazenam valores que podem ser alterados.
* Constantes: armazenam valores imutáveis.

int idade = 25; // Variável

const double Pi = 3.1415; // Constante

* Var: Permite inferência de tipo.
* Dynamic: Permite mudança de tipo em tempo de execução.
* Object: Tipo base para todos os tipos.

var nome = "João"; // O compilador infere que é string

dynamic valor = 10; // Pode mudar de tipo em tempo de execução

object qualquerCoisa = 42; // Pode armazenar qualquer coisa

3. Estruturas de Dados Simples

Arrays

int[] numeros = { 1, 2, 3, 4, 5 };

Tuples (Tuplas)

Armazena um conjunto de valores sem a necessidade de criar uma classe.

var pessoa = ("João", 30, "Engenheiro");

Console.WriteLine(pessoa.Item1); // João

4. Coleções (Collections)

Essas estruturas armazenam múltiplos valores e oferecem métodos para manipulá-los.

List (Lista Dinâmica)

List<int> lista = new List<int> { 1, 2, 3 };

lista.Add(4);

Dictionary (Dicionário)

Dictionary<string, int> idades = new Dictionary<string, int>

{

{ "Alice", 30 },

{ "Bob", 25 }

};

5. Tipos de Dados Avançados

Enums

Conjunto de valores nomeados.

enum DiasDaSemana { Domingo, Segunda, Terça, Quarta, Quinta, Sexta, Sábado };

DiasDaSemana hoje = DiasDaSemana.Segunda;

Structs

Parecido com classes, mas armazenado na stack.

struct Ponto

{

public int X;

public int Y;

}

Classes

Classes são modelos para objetos.

class Pessoa

{

public string Nome { get; set; }

public int Idade { get; set; }

}