IFRN

Tecnologia em Análise e Desenvolvimento de Sistemas

Strings e Arquivos Texto

Prof. Gilbert Azevedo

Strings

- string
  - É um tipo de dados que representa uma coleção de caracteres
  - Valores do tipo *string* são amplamente utilizados no desenvolvimento de aplicativos por representar, em geral, todos os textos que aparecem em um programa
  - Em C#, o tipo *string* é uma classe, o que facilita a sua utilização em relação ao tipo *string* de outras linguagens de programação

Declaração de String

- A declaração de variáveis *string* é feita de forma análoga à declaração de variáveis de outros tipos primitivos (int, double, ...)
  - Ex: Declaração de *strings* sem atribuição inicial
    - string s1, s2;
  - Ex: Declaração de *strings* com atribuição inicial
    - string s3 = "C++", s4 = "Algoritmos";

Operações Básicas com Strings

- Atribuição
  - O operador de atribuição "=" é utilizado para atribuir um valor a uma *string*
    - string s;
    - s = "Algoritmos";
    - s = "";
- Indexação
  - O operador de indexação "[ ]" é usado para acessar (ler) cada caractere da *string* individualmente.
  - O índice do primeiro caractere é zero.
    - string s = "Algoritmos";
    - Console.WriteLine(s[0]); // Escreve A

Operações Básicas com Strings

- Entrada de dados
  - O método ReadLine da classe Console é utilizado para ler uma string do teclado
    - string s;
    - s = Console.ReadLine();
- Saída de dados
  - O método WriteLine é usado para mostrar o conteúdo da variável string
    - string s = "Algoritmos";
    - Console.WriteLine(s);

Comparação entre Strings

- A igualdade entre *strings* é realizada através dos operadores relacionais (== e !=)
- A comparação é realizado pelo método CompareTo
  - string s1 = "G", s2 = "g";
  - Console.WriteLine(s1 == s2); // false
  - Console.WriteLine(s1 != s2); // true
  - Console.WriteLine(s1.CompareTo(s2)); // 1
  - Console.WriteLine("A".CompareTo("a")); // 1
  - Console.WriteLine("a".CompareTo("b")); // -1
  - Console.WriteLine("a".CompareTo("B")); // -1
  - Console.WriteLine("A".CompareTo("b")); // -1
  - Console.WriteLine("A".CompareTo("B")); // -1

Concatenação de Strings

- Concatenação
  - É o termo normalmente utilizado para indicar a união entre duas ou mais *strings*
  - O operador "+" é utilizado para realizar a concatenação
    - string s = "Bota";
    - s = s + "fogo"; // Atribui "Botafogo"
    - s = s + 's'; // Atribui "Botafogos"
    - s = 'a' + 'b'; // Erro: Resultado é int

Tamanho da String

- int Length;
  - Retorna o número de caracteres da *string*
  - string s = "Algoritmos";
  - int i = s.Length; // Atribui 10 a i
  - Console.WriteLine(s.Length); // Escreve 10

Método Remove

- string Remove(int startIndex);
- string Remove(int startIndex, int count);
  - Remove *count* caracteres da string, iniciando na posição *startIndex*. Se *count* é omitido, remove os caracteres a partir de *startIndex*.
  - Obs: A string original não é alterada
  - string s = "Algoritmos";
  - Console.WriteLine(s.Remove(4)); //"Algo";
  - Console.WriteLine(s.Remove(4,3)); //"Algomos"

Método Substring

- string Substring(int startIndex);
- string Substring (int startIndex, int length);
  - Retorna *length* caracteres da string, iniciando na posição *startIndex*. Se *length* é omitido, retorna os caracteres a partir de *startIndex*.
  - Obs: A string original não é alterada
  - string s = "Algoritmos";
  - Console.WriteLine(s.Substring(4)); //"ritmos";
  - Console.WriteLine(s.Substring(4,3)); //"rit"

Método IndexOf

- int IndexOf(char value);
- int IndexOf(string value);
  - Retorna a posição inicial do caractere ou string *value* dentro da string
  - Retorna -1 se *value* não for encontrado
  - string s = "Algoritmos";
  - Console.WriteLine(s.IndexOf('o')); // 3
  - Console.WriteLine(s.IndexOf("go")); // 2
  - Console.WriteLine(s.IndexOf("x")); // -1

Método Replace

- string Replace(string oldValue, string newValue);
- string Replace(char oldChar, char newChar);
  - Substitue um caractere por outro, ou uma string por outra na string original. A string original não é alterada.
  - string s = "Algoritmos";
  - string s = "Algoritmos";
  - Console.WriteLine(s.Replace('o', '0')); // Alg0ritm0s
  - Console.WriteLine(s); // Algoritmos

Método Split

- string[] Split(char [] separator)
  - Retorna um vetor de string contendo as substrings delimitadas pelo separador separator
    - string s = "10,20;30";
    - string[] v = s.Split(',', ';');
    - foreach (string ss in v)
    - Console.WriteLine(ss);
  - Escreve
    - 10
    - 20
    - 30

Arquivos Texto

- Arquivos Texto
  - É um repositório de strings armazenado no sistema de arquivos do computador, normalmente em uma pasta
  - Utiliza um padrão ou codificação para os caracteres que pode variar de acordo com o idioma utilizado
  - Normalmente utilizam a extensão “txt”
- Manipulação de Arquivos
  - O C# utiliza os tipos (classes) StreamReader e StreamWriter para controlar a leitura e escrita de dados em um arquivo
  - Ambas as classes estão definidas no espaço de nomes System.IO

Escrevendo no Arquivo

- É necessário utilizar uma variável *StreamWriter*
  - string s;
  - StreamWriter f = new StreamWriter("c:\\temp\\teste.txt", false, Encoding.Default);
  - s = Console.ReadLine();
  - while (s != "fim") {
  - f.WriteLine(s);
  - s = Console.ReadLine();
  - }
  - f.Close();

Lendo do Arquivo

- É necessário utilizar uma variável *StreamReader*
  - string s;
  - StreamReader f = new StreamReader("c:\\temp\\teste.txt", Encoding.Default);
  - s = f.ReadLine();
  - while (s != null) {
  - Console.WriteLine(s);
  - s = f.ReadLine();
  - }
  - f.Close();

Exercícios

1. Ler uma string e contar quantas palavras tem nela.
   Ex.: "Tecnologia em Análise e Desenvolvimento de Sistemas" → 7 palavras
2. Ler uma string e mostrar de trás para frente.
   Ex.: "Tecnologia em Análise e Desenvolvimento de Sistemas" → "sametsiS ed otnemivlovneseD e esilánA me aigolonceT"
3. Ler uma string e mostrar as iniciais de cada palavra.
   Ex.: "Tecnologia em Análise e Desenvolvimento de Sistemas" → "TeAeDdS"
4. Ler uma string e escrever cada palavra desta de trás para frente.
   Ex.: "Tecnologia em Análise e Desenvolvimento de Sistemas" → "aigolonceT me esilánA e otnemivlovneseD ed sametsiS"
5. Fazer uma agenda telefônica armazenando nomes e telefones em um arquivo texto