» KIỂU STRING

Name: Nguyễn Chí Hiếu
Date: 2020

» NỘI DUNG

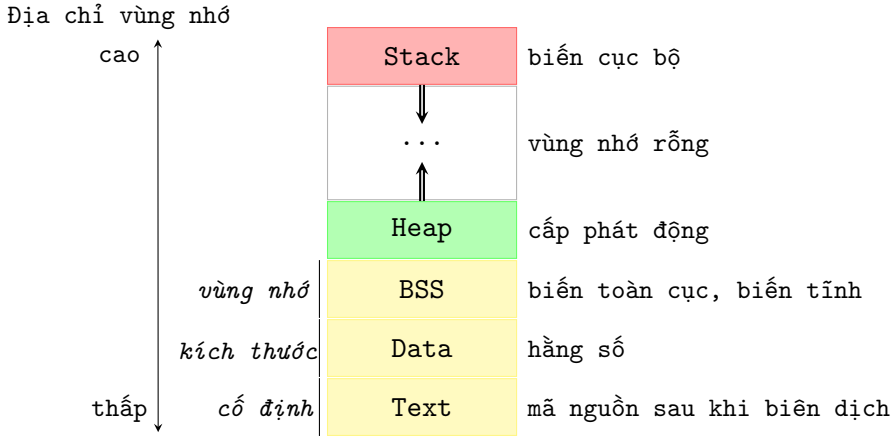
» CÁC KHÁI NIỆM CƠ BẢN

Chuỗi ký tự *(string)*

* Là một dãy các ký tự liên tiếp nhau.
* Trong C#, từ khóa string là bí danh/tên giả *(alias)* của lớp String (trong namespace System).
* String và string được sử dụng giống nhau.
* String là kiểu dữ liệu bất biến *(không thay đổi giá trị sau khi được khởi tạo trong vùng nhớ heap)*. Do đó, giá trị thay đổi thì chương trình sẽ tạo thêm một đối tượng mới trong vùng nhớ heap.

» CÁC KHÁI NIỆM CƠ BẢN

Vùng nhớ



» CÁC KHÁI NIỆM CƠ BẢN

Chuỗi ký tự *(string)*

* Biến kiểu string được lưu vùng nhớ stack còn giá trị của nó được lưu vùng nhớ heap.
* string là kiểu dữ liệu bất biến *(không thay đổi giá trị sau khi được khởi tạo trong vùng nhớ heap)*. Nếu giá trị của biến thay đổi, thì chương trình sẽ tạo thêm một đối tượng mới trong vùng nhớ heap.

» CÁC KHÁI NIỆM CƠ BẢN

Các ký tự thoát *(escape sequence)*

Ký tự	Ý nghĩa
\'	Ngoặc đơn '
\"	Ngoặc kép "
\?	Dấu ?
\\	\
\0	null
\a	Âm thanh cảnh báo
\b	Xóa 1 ký tự bên trái
\f	Sang trang
\b	Xuống dòng
\r	Trở về đầu dòng
\t	Tab theo dòng
\v	Tab theo cột

Các ký tự thoát *(escape sequence)*

```
1  Console.WriteLine("1\n2\n3\n4\n5");
```

```
> 1
> 2
> 3
> 4
> 5
```

» CÁC KHÁI NIỆM CƠ BẢN

Các ký tự thoát *(escape sequence)*

* Chuỗi đường dẫn của một tập tin hay thư mục có chứa các dấu "\". Do đó, chuỗi cần sử dụng các ký tự "\\" để hiển thị đúng.
* Ngoài ra, ký tự @ có thể được đặt trước chuỗi để thay thế các ký tự "\\".

```
1 Console.WriteLine("C:\\Users\\Hieu\\Desktop\\note.txt");
2
3 Console.WriteLine(@"C:\Users\Hieu\Desktop\note.txt");
```

```
> C:\Users\Hieu\Desktop\note.txt
> C:\Users\Hieu\Desktop\note.txt
```

» KHAI BÁO

Khai báo và khởi tạo

* Truyền giá trị trực tiếp
* Gán bằng giá trị trả về của một hàm
* Gán bằng một chuỗi kiểu char

» KHAI BÁO

Khai báo và khởi tạo

```
1  static void Main(string[] args)
2  {
3      string s1, s2, s3, s4, s5;
4      s2 = "";
5      s3 = String.Empty;
6      s4 = "Hello world";
7      s5 = GetMessage();
8      //...
9  }
10 static string GetMessage()
11 {
12     return "Hi All";
13 }
```

Khai báo và khởi tạo

Khởi tạo từ một mảng kiểu char.

```
1 char[] fruits = { 'p', 'i', 'n', 'e', 'a', 'p', 'p', 'l
     ', 'e' };
2 string s6 = fruits;
3
4 Console.WriteLine(s6);
```

» TRUY XUẤT PHẦN TỬ

Truy xuất dựa vào chỉ số

Chỉ số chuỗi ký tự là một số nguyên dương từ 0 đến n - 1, với n là chiều dài chuỗi.

* n: số dòng của mảng, gọi hàm GetLength(0) để trả về số dòng.
* m: số cột của mảng, gọi hàm GetLength(1) để trả về số cột.

```
1  string s = "sun";
2  char[0] = 'r';
3
4  Console.WriteLine(s);
5  Console.WriteLine(s[0]);
```

» TRUY XUẤT PHẦN TỬ

Truy xuất dựa vào hàm

Sử dụng hàm SubString để lấy một chuỗi con từ chuỗi ban đầu.

* SubString(i): lấy chuỗi con từ vị trí i đến cuối.
* SubString(i, j): lấy chuỗi con từ vị trí i đến j.

```
1  string s = "Hello World";
2
3  Console.WriteLine(s);
4  Console.WriteLine(s[0]);
```

» CÁC THAO TÁC VỚI CHUỖI KÝ TỰ

Chuỗi và ký tự

* Length: trả về chiều dài chuỗi ký tự
* ToCharArray(): trả về mảng các ký tự
* Split(char c): tách chuỗi khi gặp ký tự c, kết quả trả về là một mảng kiểu chuỗi

So sánh chuỗi

* Equal(string s): so sánh chuỗi hiện tại và chuỗi s
* Compare(string s1, string s2, true): có phân biệt hoa, thường
* Compare(string s1, string s2, false): không phân biệt hoa, thường

» CÁC THAO TÁC VỚI CHUỖI KÝ TỰ

Các hàm kiểm tra

* IsNullOrEmpty(): kiểm tra chuỗi rỗng
* IsDigit, IsUpper(), IsLower()
* Contains(string s): kiểm tra có chứa chuỗi s
* IndexOf(string s): nếu tìm thấy trả về vị trí của chuỗi s, ngược lại trả về -1
* IndexOf(string s, int startIndex): bắt đầu thực hiện từ vị trí startIndex, nếu tìm thấy trả về vị trí của chuỗi s, ngược lại trả về -1

» CÁC THAO TÁC VỚI CHUỖI KÝ TỰ

Các hàm cập nhật chuỗi

* Trim(): xóa khoảng trắng giữa các ký tự (nếu có)
* Concat(string s1, string s2): nối chuỗi s1 và s2
* Remove(string s): xóa chuỗi từ vị trí i đến cuối chuỗi
* Remove(int i, int j): xóa chuỗi từ vị trí i đến j
* Replace(string s1, string s2): thay thế chuỗi s1 bằng s2
* Copy(string s): tạo chuỗi mới sao chép từ chuỗi s
* Insert(int i, string s): chèn chuỗi s vào vị trí i

» CÁC THAO TÁC VỚI CHUỖI KÝ TỰ

Ví dụ

Đếm số lượng từ trong một chuỗi.

```
1  static int CountWords(string s)
2  {
3      char[] c = new char[] { ' ',  '.',  ',' };
4      string[] w = str.Split(c);
5      return w.Length;
6  }
7
8  static void Main() {
9      int count = CountWords("Learn how to write any
           application using the C# programming language on
           the .NET platform.");
10     Console.Write(count);
11 }
```

» CÁC THAO TÁC VỚI CHUỖI KÝ TỰ

Ví dụ

Đếm số từ lặp lại trong một chuỗi.

```
1  static void CountRepeatedWord(string s)
2  {
3      char[] c = new char[] { ' ',  '.',  ',' };
4      string[] w = s.Split(c);
5      string[] w_distinct = w.Distinct().ToArray();
6      int[] repeated = new int[w_distinct.Length];
7
8      for (int i = 0; i < w_distinct.Length; i++)
9      {
10         int pos = -1;
11         while ((pos = s.IndexOf(w_distinct[i], pos + 1))
               != -1)
12             repeated[i]++;
13     }
```

» CÁC THAO TÁC VỚI CHUỖI KÝ TỰ

Ví dụ

Đếm số từ lặp lại trong một chuỗi.

```
14      for (int i = 0; i < w_distinct.Length; i++)
15          Console.WriteLine("{0}: {1}", w_distinct[i],
            repeated[i]);
16  }
```

» CÁC THAO TÁC VỚI CHUỖI KÝ TỰ

Định dạng chuỗi

* Format(string s, object o): chuỗi ký tự được định dạng lại theo chuỗi đing dạng s, tham khảo nội dung về Nhập xuất dữ liệu trong ngôn ngữ C#.

» CÁC THAO TÁC VỚI CHUỖI KÝ TỰ

Tối ưu kiểu chuỗi với StringBuilder

StringBuilder tương tự String nhưng là kiểu dữ liệu có thể thay đổi giá trị trong vùng nhớ Heap.

* Thực hiện thao tác nối chuỗi nhanh và tốn ít vùng nhớ.
* Thường được sử dụng khi chuỗi được thay đổi giá trị nhiều lần.

```
1 string str = "Hello ";
2
3 str += "World!";
4
5 Console.Write(str);
```

```
1 StringBuilder str = new
     StringBuilder();
2
3 str.Append("Hello ");
4 str.Append("World!");
5
6 Console.Write(str.ToString());
```

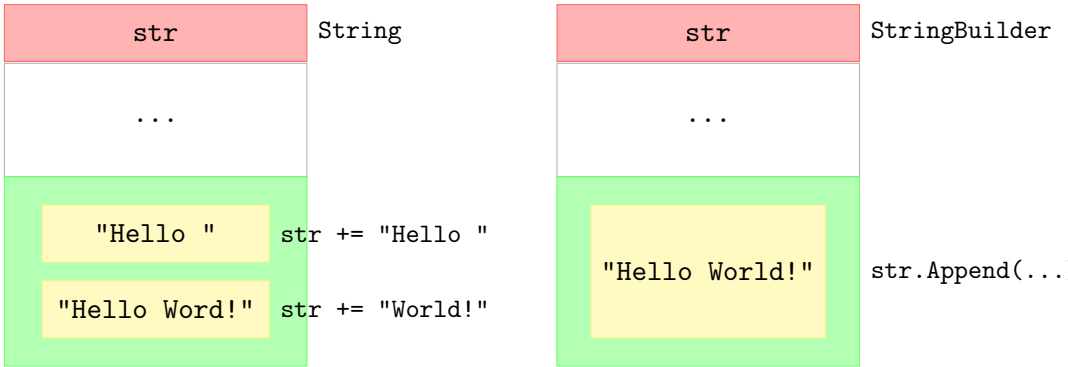
Tối ưu kiểu chuỗi với StringBuilder

StringBuilder cũng hỗ trợ một số hàm xử lý chuỗi:

* Append()
* Insert()
* Remove()
* Replace()
* ToString()

» CÁC THAO TÁC VỚI CHUỖI KÝ TỰ

Tối ưu kiểu chuỗi với StringBuilder



» BÀI TẬP

1. Nhập vào một chuỗi ký tự từ bàn phím. Viết hàm chuyển chuỗi vừa nhập thành chuỗi ký tự thường và không có khoảng trắng.
2. Nhập vào một chuỗi ký tự từ bàn phím. Viết hàm đếm xem chuỗi có bao nhiêu từ.
3. Nhập vào một chuỗi ký tự từ bàn phím. Viết hàm tìm từ dài nhất trong tất cả các từ trong chuỗi.
4. Nhập vào một tài khoản email từ bàn phím. Viết hàm kiểm tra email có hợp lệ hay không? Email hợp lệ theo định dạng email@domain.com

» BÀI TẬP

5. Mật mã Caesar là một mật mã dịch chuyển. Mỗi ký tự trong bản rõ được thay thế bằng một ký tự cách nó một đoạn trong bảng chữ cái để tạo thành bản mã. Giả sử chọn n = 3, A sẽ được thay bằng D. Viết hàm mã hóa và giải mã chuỗi ký tự nhập từ bàn phím.