**The predefined types in C# are:**

**Value types**

• Numeric

• Signed integer (sbyte, short, int, long)

• Unsigned integer (byte, ushort, uint, ulong)

• Real number (float, double, decimal)

• Logical (bool)

• Character (char)

**Reference types**

• String (string)

• Object (object)

* **ENUM**

**enum** *Name*

{

Constant\_Name\_1 = Value\_1**,**

Constant\_Name\_2 = Value\_2**,**

…**,**

}**;**

* **2 types**:

-built-in

-user-defined

* **Foreach**

**foreach** (*<Kiểu tập hợp>* *<Tên tên truy cập thành phần>* **in** *<Tên tập hơp>*)

<khối lệnh>

#region để viết tiếng Việt ra màn hình Console thì thêm dòng này trước khi xuất

//Console.OutputEncoding = Encoding.UTF8;

//Console.WriteLine("â");

//Console.WriteLine("ăn");

//int var = 3;

//Console.WriteLine(var);

//Console.WriteLine("ngủ");

#endregion

#region vẽ hình chữ nhật

//int n = 10, m = 30;

//for(int i=0; i<n; i++)

//{

// for (int j = 0; j < m; j++)

// if (i % (n - 1) == 0 || i % (n - 1) != 0 && j % (m - 1) == 0)

// Console.Write('\*');

// else

// Console.Write(' ');

// Console.WriteLine();

//}

#endregion

#region radom

//Random rand = new Random();

//int valueRandom = rand.Next(10, 100);

#endregion

#region mảng một chiều

//int[] arr1;

//int[] arr2 = new int[2] { 1,2};//phải có số lượng phần tử tối đa ->cấp phát vùng nhớ trong heap

//int[] arr3 = {1, 2};

#endregion

#region mảng hai chiều

//int[,] dou\_arr1;

//int[,] dou\_arr2 = new int[2,2] { { 0, 1 }, { 2, 3 } };//phải có số lượng phần tử tối đa ->cấp phát vùng nhớ trong heap

//int[,] dou\_arr3 = {

// { 0 , 1 },

// { 2 , 3 }

// };

//int[,] arr = {

// {1,2,3},

// {4,5,6},

// {7,8,9}

// };

//for (int i = 0; i < arr.GetLength(0); i++)//GetLength(0) lay ra chieu thu nhat

//{

// for (int j = 0; j < arr.GetLength(1); j++)//GetLength(1) lay ra chieu thu 2

// Console.Write("{0,5}", arr[i, j]);//{0,5} la dinh dang cho khoang cach giua cac phan tu kh xuat ra

// Console.WriteLine();

//}

#endregion

#region thao tác với string

//string fullName;

//string result = "";

//Console.Write("Nhap ho ten: ");

//fullName = Console.ReadLine();

////xóa các khoang trang dau va cuoi chuoi

//fullName = fullName.Trim();

////duyệt chuỗi, nếu tìm thấy 2 khoảng trắng thì thay thế = 1 khoảng trắngfirstChar

//while(fullName.IndexOf(" ") != -1)

//{

// fullName = fullName.Replace(" ", " ");

//}

////tách thành từng từ riêng biệt

//string[] SubName = fullName.Split(' ');

//for (int i = 0; i < SubName.Length; i++)

//{

// string firstChar = SubName[i].Substring(0, 1);// Substring lấy ra ký tự đầu tiên (từ 0 -> 1)

// string otherChar = SubName[i].Substring(1);// lấy ra ký tự từ vị trí thứ 1 tới hết từ

// SubName[i] = firstChar.ToUpper() + otherChar.ToLower();

// //SubName[i] = SubName[i].Substring(0, 1).ToUpper() + SubName[i].Substring(1).ToLower();

// result += SubName[i] + " ";

//}

//Console.WriteLine("Ho ten sau khi chuan hoa la: " + result);

//// ? nếu thay dấu cách bằng tab thì điều gì sẽ xảy ra

#endregion

#region Regular Expression

//dùng @ cho code dễ nhìn, có thể thay @"\d" bằng "\\d"

Regex reg = new Regex(@"\d"); // /d là số

Match result = reg.Match("-VoMinhTam 10021998-");

// cách 1

// // kết quả chỉ in ra số đầu tiên

// cách 2 dùng vòng lặp và NextMatch

//do

//{

// Console.WriteLine(result.ToString());

// result = result.NextMatch();

//} while (result != Match.Empty);//lặp tới khi nào không còn giá trị trùng với khoảng 0->9

//cách 3 dùng vòng lặp foreach

//foreach (Match item in reg.Matches("VoMinhTam 10021998"))

//{

// Console.WriteLine(item.ToString());

//}

#endregion

#region Group => xảy ra lỗi?

//Regex re = new Regex(@"?(<hours>\d+):(?<minutes>\d+):(?<seconds>\d+)");

//foreach (Match item in re.Matches("02/09/2017 19:27:06 159.696.5.3"))

//{

// Console.WriteLine("Match: " + item.ToString());

// Console.WriteLine("Hours: " + item.Groups["hours"]);

// Console.WriteLine("Minutes: " + item.Groups["minutes"]);

// Console.WriteLine("Seconds: " + item.Groups["seconds"]);

//}

#endregion

#region Capture 1

//Regex Re = new Regex(@"(?<times>(\d|:)+)\s" + @"(?<companies>\S+)\s" + @"(?<ip>(\d|\.)+)\s" + @"(?<companies>\S+)");

//// lưu ý: dấu . là \.

//foreach (Match item in Re.Matches("09:06:03 VMT 195.386.28.4 Apple"))

//{

// Console.WriteLine("Time: " + item.Groups["times"]);

// Console.WriteLine("Company: " + item.Groups["companies"]);

// Console.WriteLine("IP: " + item.Groups["ip"]);

// Console.WriteLine("Company: " + item.Groups["companies"]);

//}

#endregion

#region Capture 2

Regex Re = new Regex(@"(?<times>(\d|:)+)\s" + @"(?<companies>\S+)\s" + @"(?<ip>(\d|\.)+)\s" + @"(?<companies>\S+)");

// lưu ý: dấu . là \.

foreach (Match item in Re.Matches("09:06:03 VMT 195.386.28.4 Apple"))

{

Console.WriteLine("Time: " + item.Groups["times"]);

Console.WriteLine("IP: " + item.Groups["ip"]);

Console.Write("Company: ");

foreach (Capture i in item.Groups["companies"].Captures)

{

Console.Write(i.ToString() + "\t");

}

}

#endregion