C# kolekcije

# Dictionary <Tkey, Tvalue>

using System;

using System.Collections.Generic;

public class Example

{

public static void Main()

{

// Create a new dictionary of strings, with string keys.

//

Dictionary<string, string> openWith =

new Dictionary<string, string>();

// Add some elements to the dictionary. There are no

// duplicate keys, but some of the values are duplicates.

openWith.Add("txt", "notepad.exe");

openWith.Add("bmp", "paint.exe");

openWith.Add("dib", "paint.exe");

openWith.Add("rtf", "wordpad.exe");

// The Add method throws an exception if the new key is

// already in the dictionary.

try

{

openWith.Add("txt", "winword.exe");

}

catch (ArgumentException)

{

Console.WriteLine("An element with Key = \"txt\" already exists.");

}

// The Item property is another name for the indexer, so you

// can omit its name when accessing elements.

Console.WriteLine("For key = \"rtf\", value = {0}.",

openWith["rtf"]);

// The indexer can be used to change the value associated

// with a key.

openWith["rtf"] = "winword.exe";

Console.WriteLine("For key = \"rtf\", value = {0}.",

openWith["rtf"]);

// If a key does not exist, setting the indexer for that key

// adds a new key/value pair.

openWith["doc"] = "winword.exe";

// The indexer throws an exception if the requested key is

// not in the dictionary.

try

{

Console.WriteLine("For key = \"tif\", value = {0}.",

openWith["tif"]);

}

catch (KeyNotFoundException)

{

Console.WriteLine("Key = \"tif\" is not found.");

}

// When a program often has to try keys that turn out not to

// be in the dictionary, TryGetValue can be a more efficient

// way to retrieve values.

string value = "";

if (openWith.TryGetValue("tif", out value))

{

Console.WriteLine("For key = \"tif\", value = {0}.", value);

}

else

{

Console.WriteLine("Key = \"tif\" is not found.");

}

// ContainsKey can be used to test keys before inserting

// them.

if (!openWith.ContainsKey("ht"))

{

openWith.Add("ht", "hypertrm.exe");

Console.WriteLine("Value added for key = \"ht\": {0}",

openWith["ht"]);

}

// When you use foreach to enumerate dictionary elements,

// the elements are retrieved as KeyValuePair objects.

Console.WriteLine();

foreach( KeyValuePair<string, string> kvp in openWith )

{

Console.WriteLine("Key = {0}, Value = {1}",

kvp.Key, kvp.Value);

}

// To get the values alone, use the Values property.

Dictionary<string, string>.ValueCollection valueColl =

openWith.Values;

// The elements of the ValueCollection are strongly typed

// with the type that was specified for dictionary values.

Console.WriteLine();

foreach( string s in valueColl )

{

Console.WriteLine("Value = {0}", s);

}

// To get the keys alone, use the Keys property.

Dictionary<string, string>.KeyCollection keyColl =

openWith.Keys;

// The elements of the KeyCollection are strongly typed

// with the type that was specified for dictionary keys.

Console.WriteLine();

foreach( string s in keyColl )

{

Console.WriteLine("Key = {0}", s);

}

// Use the Remove method to remove a key/value pair.

Console.WriteLine("\nRemove(\"doc\")");

openWith.Remove("doc");

if (!openWith.ContainsKey("doc"))

{

Console.WriteLine("Key \"doc\" is not found.");

}

}

}

# List <T>

using System;

using System.Collections.Generic;

public class Example

{

public static void Main()

{

List<string> dinosaurs = new List<string>();

Console.WriteLine("\nCapacity: {0}", dinosaurs.Capacity);

dinosaurs.Add("Tyrannosaurus");

dinosaurs.Add("Amargasaurus");

dinosaurs.Add("Mamenchisaurus");

dinosaurs.Add("Deinonychus");

dinosaurs.Add("Compsognathus");

Console.WriteLine();

foreach(string dinosaur in dinosaurs)

{

Console.WriteLine(dinosaur);

}

Console.WriteLine("\nCapacity: {0}", dinosaurs.Capacity);

Console.WriteLine("Count: {0}", dinosaurs.Count);

Console.WriteLine("\nContains(\"Deinonychus\"): {0}",

dinosaurs.Contains("Deinonychus"));

Console.WriteLine("\nInsert(2, \"Compsognathus\")");

dinosaurs.Insert(2, "Compsognathus");

Console.WriteLine();

foreach(string dinosaur in dinosaurs)

{

Console.WriteLine(dinosaur);

}

// Shows accessing the list using the Item property.

Console.WriteLine("\ndinosaurs[3]: {0}", dinosaurs[3]);

Console.WriteLine("\nRemove(\"Compsognathus\")");

dinosaurs.Remove("Compsognathus");

Console.WriteLine();

foreach(string dinosaur in dinosaurs)

{

Console.WriteLine(dinosaur);

}

dinosaurs.TrimExcess();

Console.WriteLine("\nTrimExcess()");

Console.WriteLine("Capacity: {0}", dinosaurs.Capacity);

Console.WriteLine("Count: {0}", dinosaurs.Count);

dinosaurs.Clear();

Console.WriteLine("\nClear()");

Console.WriteLine("Capacity: {0}", dinosaurs.Capacity);

Console.WriteLine("Count: {0}", dinosaurs.Count);

}

}

# ArrayList

using System;

using System.Collections;

public class CSharpApp

{

class Empty

{}

static void Main()

{

ArrayList da = new ArrayList();

da.Add("Visual Basic");

da.Add(344);

da.Add(55);

da.Add(new Empty());

da.Remove(55);

foreach(object el in da)

{

Console.WriteLine(el);

}

}

}

# Hashtable

using System;

using System.Collections;

namespace CollectionsApplication

{

class Program

{

static void Main(string[] args)

{

Hashtable ht = new Hashtable();

ht.Add("001", "Zara Ali");

ht.Add("002", "Abida Rehman");

ht.Add("003", "Joe Holzner");

ht.Add("004", "Mausam Benazir Nur");

ht.Add("005", "M. Amlan");

ht.Add("006", "M. Arif");

ht.Add("007", "Ritesh Saikia");

if (ht.ContainsValue("Nuha Ali"))

{

Console.WriteLine("This student name is already in the list");

}

else

{

ht.Add("008", "Nuha Ali");

}

// Get a collection of the keys.

ICollection key = ht.Keys;

foreach (string k in key)

{

Console.WriteLine(k + ": " + ht[k]);

}

Console.ReadKey();

}

}

}