Struct record

{id

Name

Address

}

Id name address

Enum link

<https://www.c-sharpcorner.com/UploadFile/puranindia/enum-in-CSharp/>

Generics link

<https://www.c-sharpcorner.com/UploadFile/84c85b/using-generics-with-C-Sharp/>

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp2

{

class Program

{

static void Main(string[] args)

{

// examples

// ArrayList

// STack Queue HashSet

// The advantage of Arrays is type safe, strongly typed

// it allosw one type

// The above collections are non generic

ArrayList list = new ArrayList();

list.Add(1);

list.Add(1);

list.Add(1);

list.Add(1);

list.Add("1");

// Here it does boxing

// converts value type to reference type

foreach (var x in list) // Here it is doing unboxing

{

Console.WriteLine(x);

}

// We need to take advantages of collection and arrays both

// Generic collections

// List<type>

// Stack<>

List<int> list1 = new List<int>();

list1.Add(1);

}

}

}

Regex : Regular Expression

For pattern matching

<https://www.geeksforgeeks.org/what-is-regular-expression-in-c-sharp/>

Parameterized Query??

App.config file : we can store connectionstring there

Threads

Single Tasking : One task at a time

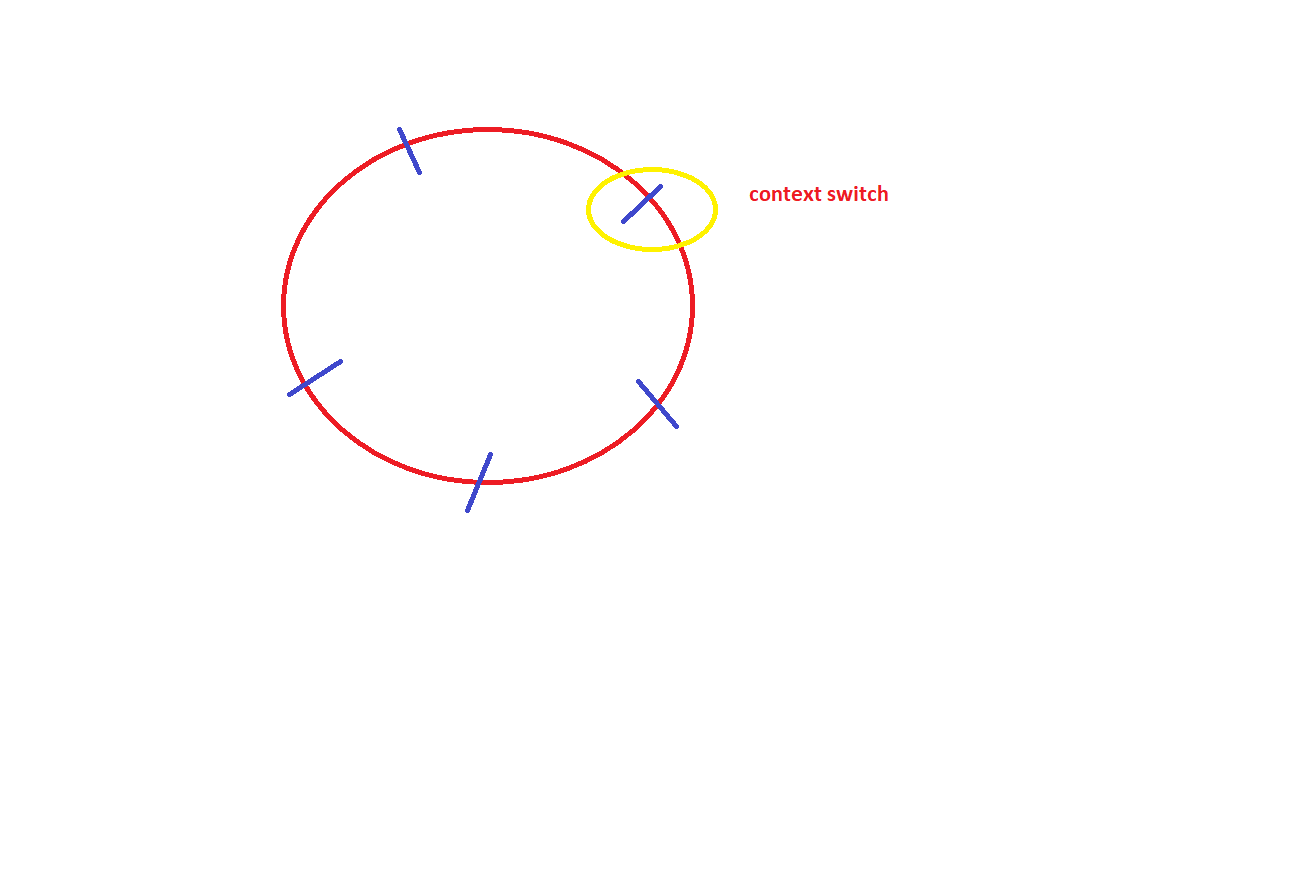
Multi Tasking : Multiple tasks together

1. Multiprogramming (different programs are running together)
2. Multithreading : different statements of one program running together

Context switch :

In Single tasking : CPU complete time allocated to one task

In Multiprogramming : CPU divides its time to different tasks

Context switch time is less in Multithreading

using System;

using System.Threading;

class ThreadDemo

{

public void display1()

{

for (int i = 1; i <= 10; i++)

{

Console.WriteLine(i);

}

}

public void display2()

{

for (int i = 11; i <= 20; i++)

{

Console.WriteLine(i);

}

}

static void Main()

{

ThreadDemo program = new ThreadDemo();

Thread t1 = new Thread(program.display1);

Thread t2 = new Thread(program.display2);

t1.Start();

t2.Start();

}

}

using System;

using System.Threading;

class ThreadDemo

{

public void display1()

{

for (int i = 1; i <= 10; i++)

{

Console.WriteLine(i);

Thread.Sleep(100);

}

}

public void display2()

{

for (int i = 11; i <= 20; i++)

{

Console.WriteLine(i);

}

}

static void Main()

{

ThreadDemo program = new ThreadDemo();

Thread t1 = new Thread(program.display1);

Thread t2 = new Thread(program.display2);

t1.Start();

t1.Join();

t2.Start();

}

}

File Handling : you want to work with files

Text files , binary files

using System;

using System.IO;

namespace ConsoleApp2

{

class DirectoryDemo

{

static void Main()

{

DriveInfo[] di = DriveInfo.GetDrives();

Console.WriteLine("Total Partitions");

foreach (DriveInfo items in di)

{

Console.WriteLine(items.Name);

}

//DirectoryInfo directoryInfos = new DirectoryInfo(@"E:\Trainings");

//FileInfo[] files = directoryInfos.GetFiles();

//foreach(FileInfo f in files)

// Console.WriteLine(f.FullName + " " + f.CreationTime);

DirectoryInfo directoryInfos1 = new DirectoryInfo(@"E:\Trainings");

DirectoryInfo[] directories1 = directoryInfos1.GetDirectories();

foreach (DirectoryInfo f in directories1)

Console.WriteLine(f.FullName + " " + f.CreationTime);

}

}

}

using System;

using System.IO;

namespace ConsoleApp2

{

class DirectoryDemo

{

static void Main()

{

// File.Create(@"E:\Trainings\newfile.txt");

File.WriteAllText(@"E:\Trainings\newfile.txt", "Hello All");

File.AppendAllText(@"E:\Trainings\newfile.txt", "Second line");

File.Copy(@"E:\Trainings\newfile.txt", @"E:\Trainings\newfile2.txt");

File.Delete(@"E:\Trainings\newfile.txt");

string[] lines = new string[]

{ "first", "second", "third"};

File.AppendAllLines(@"E:\Trainings\newfile.txt",lines);

FileInfo fileInfo = new FileInfo(@"E:\Trainings\newfile2.txt");

fileInfo.AppendText();

fileInfo.CreateText();

}

}

}

using System;

using System.IO;

namespace ConsoleApp2

{

class DirectoryDemo

{

static void Main()

{

FileStream fs = new FileStream(@"E:\Trainings\newcopy.txt",

FileMode.Create,FileAccess.Write);

StreamWriter writer = new StreamWriter(fs);

string ch = "y";

while (ch == "y")

{

Console.WriteLine("Enter Name");

string name = Console.ReadLine();

writer.Write(name);

Console.WriteLine("Enter other Name ?");

ch = Console.ReadLine();

}

writer.Close();

fs.Close();

}

}

}

FileStream fs = new FileStream(@"E:\Trainings\newcopy.txt",

FileMode.Append,FileAccess.Write);

using System;

using System.IO;

namespace ConsoleApp2

{

class DirectoryDemo

{

static void Main()

{

//FileStream fs = new FileStream(@"E:\Trainings\newcopy.txt",

// FileMode.Append,FileAccess.Write);

//StreamWriter writer = new StreamWriter(fs);

//string ch = "y";

//while (ch == "y")

//{

// Console.WriteLine("Enter Name");

// string name = Console.ReadLine();

// writer.Write(name);

// Console.WriteLine("Enter other Name ?");

// ch = Console.ReadLine();

//}

//writer.Close();

//fs.Close();

FileStream fs = new FileStream(@"E:\Trainings\newcopy.txt", FileMode.Open, FileAccess.Read);

StreamReader reader = new StreamReader(fs);

int n;

while ((n= reader.Read()) !=-1)

{

Console.Write((char)n);

}

reader.Close();

fs.Close();

}

}

}

using System;

using System.IO;

namespace ConsoleApp2

{

class DirectoryDemo

{

static void Main()

{

//FileStream fs = new FileStream(@"E:\Trainings\new1.txt",

// FileMode.CreateNew, FileAccess.Write);

//StreamWriter writer = new StreamWriter(fs);

//string ch = "y";

//while (ch == "y")

//{

// Console.WriteLine("Enter Name");

// string name = Console.ReadLine();

// writer.WriteLine(name);

// Console.WriteLine("Enter other Name ?");

// ch = Console.ReadLine();

//}

//writer.Close();

//fs.Close();

FileStream fs = new FileStream(@"E:\Trainings\new1.txt", FileMode.Open, FileAccess.Read);

StreamReader reader = new StreamReader(fs);

string n;

while ((n= reader.ReadLine()) !=null)

{

Console.WriteLine(n);

}

reader.Close();

fs.Close();

}

}

}

using System;

using System.IO;

namespace ConsoleApp2

{

class DirectoryDemo

{

static void Main()

{

FileStream fs = new FileStream(@"E:\Trainings\new2.txt",

FileMode.CreateNew, FileAccess.Write);

BinaryWriter writer = new BinaryWriter(fs);

string ch = "y";

while (ch == "y")

{

Console.WriteLine("Enter Name");

string name = Console.ReadLine();

writer.Write(name);

Console.WriteLine("Enter other Name ?");

ch = Console.ReadLine();

}

writer.Close();

fs.Close();

fs = new FileStream(@"E:\Trainings\new2.txt", FileMode.Open, FileAccess.Read);

BinaryReader reader = new BinaryReader(fs);

int n;

while ((n= reader.Read()) !=-1)

{

Console.Write((char)n);

}

reader.Close();

fs.Close();

}

}

}

<https://www.c-sharpcorner.com/UploadFile/mahesh/how-to-read-a-text-file-in-C-Sharp/>