

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Net;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading;

using System.Threading.Tasks;

namespace WebParsing

{

class Program

{

static MemoryStream ms=new MemoryStream();

static StringBuilder sbBase=new StringBuilder();

static void Main(string[] args)

{

WebRequest webRequest = WebRequest.Create("http://www.mail.ru");

WebResponse webResponse = webRequest.GetResponse();

var stream = webResponse.GetResponseStream();

var sr = new StreamReader(stream);

new Thread(ReadStr).Start(sr);

Console.ReadKey();

ms.Close();

}

public static void ReadStr(object s)

{

var ss = s as StreamReader;

Encoding enc = Encoding.Unicode;

char[] buf = new char[100];

var sw = File.OpenWrite("test.txt");

while (!ss.EndOfStream)

{

sbBase.AppendLine(ss.ReadLine());

ss.Read(buf, 0, buf.Length);

sw.Write(enc.GetBytes(buf), 0, buf.Length);

ms.Write(enc.GetBytes(buf), 0, buf.Length);

}

ss.Close();

sw.Close();

Parse();

Console.WriteLine("End");

}

static public void Parse()

{

StringBuilder sb = new StringBuilder();

ms.Position = 0;

var sr = new StreamReader(ms, Encoding.Unicode);

while (!sr.EndOfStream)

{

sb.AppendLine(sr.ReadLine());

}

sr.Close();

//Console.WriteLine(sb.ToString());//@"href='(?<link>\S+)'>"

NewParse(@"(?<link>[a-zA-z0-9\_-]+@[a-z]+\.[a-z]{2,5})",sb);//@"src='(?<link>[^']+)'"

NewParse(@"href=.(?<link>\S+).", sb);

ms = new MemoryStream(ms.ToArray());

}

public static void NewParse(string s,StringBuilder sb)

{

Regex reg = new Regex(s);

var coll = reg.Matches(sb.ToString());

foreach (Match item in coll)

{

string i1 = item.Groups["link"].Value;

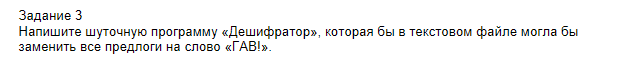
Console.WriteLine(i1);

}

}

}

}

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

namespace DeshefratorGAV

{

class Program

{

static void Main(string[] args)

{

var fr = File.OpenRead("1.txt");

Regex reg=new Regex(@"\s[а-яА-ЯA-Z]{3}");

var sr = new StreamReader(fr, Encoding.Default);

string str = sr.ReadToEnd();

sr.Close();

string tmp=reg.Replace(str, Evaluator);

Console.WriteLine(tmp);

var sw=new StreamWriter(File.OpenWrite("1.txt"),Encoding.Default);

sw.WriteLine(tmp);

sw.Close();

// File.WriteAllText("1.txt",tmp);

Console.ReadKey();

}

private static string Evaluator(Match match)

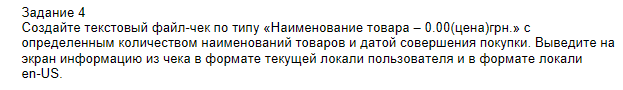
{

return "AAA";

}

}

}

using System;

using System.Collections.Generic;

using System.Globalization;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

namespace CheackTovara

{

class Program

{

public class Cheks:IFormattable

{

public int Sale { get; set; }

public String Product { get; set; }

public string ToString(string format, IFormatProvider formatProvider)

{

if (String.IsNullOrEmpty(format))

format = "baks";

if (formatProvider == null)

formatProvider = CultureInfo.CurrentCulture;

switch (format)

{

case "GRN":

return string.Format("Наименование товара - {0,5}; Цена - {1,5} ",Product,Sale.ToString("C2",formatProvider));

case "DOL":

return string.Format("Наименование товара - {0,5}; Цена - {1,5} ", Product, Sale.ToString("C2", formatProvider));

case "RUB":

return string.Format("Наименование товара - {0,5}; Цена - {1,5} ", Product, Sale.ToString("C2", formatProvider));

default:

throw new ArgumentException("not find format");

}

}

}

static void Main(string[] args)

{

CultureInfo ci = CultureInfo.CurrentCulture;

var us = CultureInfo.GetCultureInfo("en-US");

var uk = CultureInfo.GetCultureInfo("uk-UA");

var chek = new Cheks() {Sale = 1000,Product = "Computer"};

Console.WriteLine(chek.ToString("GRN",uk));

Console.WriteLine(chek.ToString("RUB",null));

Console.WriteLine(chek.ToString("DOL", us));

Console.ReadKey();

}

}

}