

ДОДАТОК А - ЛІСТИНГ МОДУЛЯ ДЛЯ ОТРИМАННЯ ПОЧАТКОВИХ ДАНИХ ДЛЯ КЛАСТЕРИЗАЦІЇ З ВИКОРИСТАННЯМ ЗОВНІШЬОГО АРІ

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
// Підключаємо бібліотеку для роботи з файлами.
using System.IO;

namespace internet_res_search
{
    public partial class Form1: Form
    {
        // Порожня змінна назви файлу.
        string FName4Save = string.Empty;
        // Змінна про стан збереження файлу.
        bool NotSaved = false;

        public Form1 ()
        {
            InitializeComponent ();
            wb.StatusTextChanged += new EventHandler (wb_StatusTextChanged);
        }

        // Функція для відображення змін в statusbar. Подіюно Iexplorer.
        void wb_StatusTextChanged (object sender, EventArgs e)
        {
            toolStripStatusLabel1.Text = wb.StatusText;
        }

        // Функція для скидання файлів cookie.
        private void ClearCookie ()
        {
            string cPath =
System.Environment.GetFolderPath (Environment.SpecialFolder.Cookies);
            System.IO.DirectoryInfo di = new
System.IO.DirectoryInfo (System.Environment.GetFolderPath (Environment.SpecialFolder.Cookies))
;
            foreach (FileInfo file in di.GetFiles ("*. txt"))
            {
                file.Delete ();
            }
        }

        private void Form1_Load (object sender, EventArgs e)
        {
            // Викликаємо функцію скидання cookie.
            ClearCookie ();
        }

        private void Form1_FormClosing (object sender, FormClosingEventArgs e)
        {
            if (NotSaved)

```

```

{
    DialogResult dr = MessageBox.Show ("Є не збережені дані, Зберегти?",
        "Увага!", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Warning);
    if (dr == DialogResult.Yes)
    {
        // Викликаємо функцію збереження файлу.
        Save ();
    }
    else if (dr == DialogResult.No)
    {
        e.Cancel = false;
    }
    else if (dr == DialogResult.Cancel)
    {
        e.Cancel = true;
    }
}
}

// Функція для збереження файлу.
private void Save ()
{
    saveFileDialog.FileName = FName4Save;
    saveFileDialog.DefaultExt = ".txt";
    saveFileDialog.Filter = "Text files (*.txt) | *.txt | All files (*.*) | *.*";
    if (saveFileDialog.ShowDialog () == DialogResult.OK)
    {
        StreamWriter w = new StreamWriter (saveFileDialog.FileName, false,
Encoding.Default);
        foreach (string str in textBox3.Lines)
        {
            // Записуємо рядок в файл.
            w.WriteLine (str);
        }
        w.Close ();
        NotSaved = false;
    }
}

// Функція для обробки події html-документ завантажений.
private void wb_DocumentCompleted (object sender,
WebBrowserDocumentCompletedEventArgs e)
{
    // Statusbar - індикатор "Готово"
    toolStripStatusLabel1.Text = "Готово";
    // Записуємо URL-сторінки на якій знаходиться WebBrowser.
    textBox1.Text = wb.Url.ToString ();
    // Отримуємо індекс сторінки в пошуковій системі за допомогою функції html_page_index ().
    int current_page_idx = Convert.ToInt16 (html_page_index (wb.Url.ToString ()) [0]);
    // Отримуємо кількість сторінок, необхідне для сканування.
    int user_idx = Convert.ToInt16 (textBoxEnters.Text);
    // Витягуємо URL-сторінки на якій знаходиться WebBrowser.
    string wb_str = wb.Url.ToString ();
    bool t = false;
    // Перевірка URL і номер сторінки.
    if (wb_str.StartsWith ("http://rada.ua/yandsearch?") && (current_page_idx <
user_idx))
    {
        // Невелика пауза.
        System.Threading.Thread.Sleep (1000);
        // Оголошуємо змінну html-документ.

```

```

HtmlDocument htmlDoc = wb.Document;
    // Объявляем змінну HtmlElementCollection.
    HtmlElementCollection hec = htmlDoc.GetElementsByTagName ("a");
    // Проводимо аналіз змісту htmlElement-ів.
    foreach (HtmlElement he in hec)
    {
        // Перевірка на наявність гіперпосилань в результаті пошуку.
        if ((he.GetAttribute ("href"). ToString (). Length > 0) &&
(He.OuterHtml.Contains ("b-serp-item__title-link")))
        {
            // Змінна для усунення повторних гіперссилко.
            // Перевіряємо всі вже наявні рядки в списку на предмет збігу
з новою гіперпосиланням.
            foreach (string str in textBox3.Lines)
                if ("-" + he.GetAttribute ("href"). ToString () + ";" == str) t =
true;

            // Якщо не було співпадіння додаємо нове гіперпосилання в наш список.
            if (t == false)
            {
                textBox3.AppendText ("-" +
he.GetAttribute ("href"). ToString () + ";");
                // Переходимо на наступний рядок.
                textBox3.Text += Environment.NewLine;
            }
        }
    }
    // Викликаємо функцію переходу на наступну сторінку.
    next_page ();
}

// Функція для визначення індексу поточної сторінки і URL наступної.
private object [] html_page_index (string http_str)
{
    // Проводимо пошук номера сторінки пошуку.
    if (http_str.IndexOf ("? p =") < 0)
    {
        http_str = "http://rada.ua/yandsearch?p=0&text=" + textBox2.Text +
"& Lr = 213";
    }
    // Зчитування початкової позиції номера поточної сторінки.
    int pg_index = http_str.IndexOf ("? p =") + 3;
    // Отримуємо другий шматок URL.
    string http_str_cut = http_str.Substring (pg_index, http_str.Length - pg_index);
    // Зчитування кінцевої позиції номера поточної сторінки.
    int pg_index_len = http_str_cut.IndexOf ('&');
    // Отримуємо номер сторінки пошуку у вигляді рядка.
    string html_page_index_str = http_str.Substring (pg_index, pg_index_len);
    // Перетворимо рядок в число.
    int html_page_index = Convert.ToInt16 (html_page_index_str);
    // Формуємо URL следующей сторінки пошуку.
    string http_new = http_str.Substring (0, pg_index) + (html_page_index +
1).ToString () + http_str_cut.Substring (pg_index_len);
    // Записуємо індекс поточної (число int) і URL наступної сторінки (рядок string)
в масив типу object [].
    object [] idx = {html_page_index, http_new};
    // Виводимо результат функції.
    return idx;
}

// Обробчик для натискання кнопки "Пошук"
private void startButton_Click (object sender, EventArgs e)

```

```

{
    NotSaved = true;
    // Обрано перший метод навігації по пошуку (з імітацією).
    if (radioButtonMethod1.Checked)
    {
        HtmlElementCollection input_tag_collection =
wb.Document.GetElementsByTagName ("input");
        // Знаходимо htmlElement пошукового рядка.
        foreach (HtmlElement input in input_tag_collection)
        {
            if ((input.Name == "text"))
            {
                // Записуємо шукане вираження з textBox2 в htmlElement пошукової
рядки.
                input.InnerText = textBox2.Text;
            }
        }
        // Знаходимо htmlElement кнопки знайти.
        foreach (HtmlElement input in input_tag_collection)
        {
            if (input.OuterHtml.Contains ("b-form-button__input") &&
input.GetAttribute ("tabIndex") == "2")
                /* Раніше можна було по іншому визначити цю кнопку */
                /* (input.GetAttribute ("value"). ToString () == "Знайти") */
            {
                // Імітуємо натискання кнопки "Знайти".
                input.InvokeMember ("click");
            }
        }
    }
    // Обрано другий метод навігації із застосуванням URL.
    else if (radioButtonMethod2.Checked)
    {
        // В URL-рядки вже є шукане слово з textBox2.
        wb.Navigate ("http://rada.ua/yandsearch?text=" + textBox2.Text + "& lr = 213");
    }
}

// Функція для переходу на наступну сторінку пошуку.
void next_page ()
{
    // Оголошуємо змінну html-документ.
    HtmlDocument htmlDoc = wb.Document;
    // Витягуємо URL-сторінки на якій знаходиться WebBrowser.
// Перехід на наступну сторінку пошуку без натискання на гіперпосилання "next
page ".
    wb.Navigate (next_URL);
}

// Обробочик кнопки "зберегти файл txt".
private void savebutton_Click (object sender, EventArgs e)
{
    // Викликаємо функцію збереження файлу.
    Save ();
}
}
}

```

ДОДАТОК Б - ЛІСТИНГ МОДУЛЯ ОТРИМАННЯ ІНФОРМАЦІЇ ПРО ДЕПУТАТІВ

```
public class Deputat {
    protected String name;
    protected String surname;
    public void setName(String newName) {
        name = newName;
    }
    public String getName() {
        return name;
    }
    public void setSurname(String newSurname) {
        name = newSurname;
    }
    public String getSurname() {
        return surname;
    }
}

// успадковуємо клас Deputat
public class Employee extends Deputat {
    private String position;
    // створюємо конструктор
    public Employee(String n, String s, String p) {
        name = n;
        surname = s;
        position = p;
    }
    public void setPosition(String newProfession) {
        position = newProfession;
    }
    public String getPosition() {
        return position;
    }
}

public class Employee extends Deputat {
    private String position;
    private IdCard iCard;
    public Employee(String n, String s, String p) {
        name = n;
        surname = s;
        position = p;
```

```

    }
    public void setPosition(String newPosition){
        position = newPosition;
    }
    public String getPosition(){
        return position;
    }
    public void setIdCard(IdCard c){
        iCard = c;
    }
    public IdCard getIdCard(){
        return iCard;
    }
}
public class IdCard{
    private Date dateExpire;
    private int number;
    public IdCard(int n){
        number = n;
    }
    public void setNumber(int newNumber){
        number = newNumber;
    }
    public int getNumber(){
        return number;
    }
    public void setDateExpire(Date newDateExpire){
        dateExpire = newDateExpire;
    }
    public Date getDateExpire(){
        return dateExpire;
    }
}
IdCard card = new IdCard (123);
card.setDateExpire (new SimpleDateFormat ("yyyy-MM-dd"). parse ("2015-12-31"));
sysEngineer.setIdCard (card);
System.out.println (sysEngineer.getName () + "входить в парламентську групу" + sysEngineer.getPosition
());
System.out.println ("Балатувався у" + new SimpleDateFormat ("yyyy-MM-dd"). Format
(sysEngineer.getIdCard (). getDateExpire ());
public class Room{
    private int number;

```

```

public Room(int n){
    number = n;
}
public void setNumber(int newNumber){
    number = newNumber;
}
public int getNumber(){
    return number;
}
}
...
private Set room = new HashSet();
...
public void setRoom(Room newRoom){
    room.add(newRoom);
}
public Set getRoom(){
    return room;
}
public void deleteRoom(Room r){
    room.remove(r);
}
...
public class Department {
    private String name;
    private Set employees = new HashSet ();
    public Department (String n) {
        name = n;
    }
    public void setName (String newName) {
        name = newName;
    }
    public String getName () {
        return name;
    }
    public void addEmployee (Employee newEmployee) {
        employees.add (newEmployee);
        // Пов'язуємо депутата з фракцією
        newEmployee.setDepartment (this);
    }
    public Set getEmployees () {
        return employees;
    }
}

```

```

    }
    public void removeEmployee (Employee e) {
        employees.remove (e);
    }
}
...
private Department department;
...
public void setDepartment(Department d){
    department = d;
}
public Department getDepartment(){
    return department;
}
private class PastPosition {
    private String name;
    private Department department;
    public PastPosition(String position, Department dep){
        name = position;
        department = dep;
    }
    public void setName(String newName){
        name = newName;
    }
    public String getName(){
        return name;
    }
    public void setDepartment(Department d){
        department = d;
    }
    public Department getDepartment(){
        return department;
    }
}
...
private Set pastPosition = new HashSet();
...
public void setPastPosition(PastPosition p){
    pastPosition.add(p);
}
public Set getPastPosition(){
    return pastPosition;
}

```



```
}  
public void deletePastPosition(PastPosition p){  
    pastPosition.remove(p);  
}  
...  
public class Menu {  
    private static int i = 0;  
    public static void showEmployees (Employee [] employees) {  
        System.out.println ("Список співробітників:");  
        for (i = 0; i < employees.length; i ++ ) {  
            if (employees [i] instanceof Employee) {  
                System.out.println (employees [i] .getName () + "-" + employees [i] .getPosition ());  
            }  
        }  
    }  
}  
}
```

ДОДАТОК В - ЛІСТИНГ ОСНОВНИХ МОДУЛІВ СИСТЕМИ

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
//для datetime
using System.Globalization;
using System.IO;
// для DOM-моделі
using HtmlAgilityPack;
// для webclient
using System.Net;

namespace AS_KIPR
{
    public partial class FormChild : Form
    {
        string FName4Save = string.Empty;
        bool NotSaved = false;
        object[,] htmlNodesArray = new object[4][];
        int page_id;
        int HTML_element_id;
        string tag_value;
        Label page_id_Label = new Label();

        private MainForm mainForm;
        private string serverName;
        private string LoginValue;
        private string PasswValue;
        private string status;
        private int pageCounter = 0;

        DataTable tableTermDinamic = new DataTable();
        DataTable tableTermStatic = new DataTable();

        public FormChild(MainForm Mform, string server, string login, string
passw, string stat)
        {
            InitializeComponent();
            mainForm = Mform;
            serverName = server;
            LoginValue = login;
            PasswValue = passw;
            status = stat;
            wb.StatusTextChanged += new EventHandler(wb_StatusTextChanged);
            URL_listBox.SelectionMode = SelectionMode.One;
            wb2.StatusTextChanged += new EventHandler(wb2_StatusTextChanged);
            tableTermDinamic.Columns.Add("Term");
            tableTermDinamic.Columns.Add("Wave");
            tableTermStatic.Columns.Add("Term");
        }

        void wb_StatusTextChanged(object sender, EventArgs e)
        {
            tbSbarText.Text = wb.StatusText;
        }
    }
}

```

```

}
void wb2_StatusTextChanged(object sender, EventArgs e)
{
    tbSbarText2.Text = wb2.StatusText;
}

private void FormChild_FormClosed(object sender, FormClosedEventArgs e)
{
    mainForm.Dispose();
}

private void FormChild_Load(object sender, EventArgs e)
{
    mainForm.Hide();
    ClearCookie();
}

private void buttonNavigate_Click(object sender, EventArgs e)
{
    pageCounter = 0;
    URLNavigate(pageCounter);
}

private void buttonNextPage_Click(object sender, EventArgs e)
{
    pageCounter = pageCounter + 1;
    URLNavigate(pageCounter);
}

private void URLNavigate(int pageNum)
{
    string rada_serpURL = "http://rada.ua/yandsearch?lr=213&text=";
    if (pageNum == 0)
    {
        rada_serpURL = rada_serpURL + tbURL.Text;
    }
    else
    {
        rada_serpURL = rada_serpURL + tbURL.Text + "&p=" +
pageNum.ToString();
    }
    DataRow DR = dataSetDB.DataTableSearchTerms.NewRow();
    DR["Term"] = tbURL.Text;
    DR["TermDT"] = DateTime.Now;
    DR["respID"] = 1;
    DR["DTHour"] = DateTime.Now.Hour;
    dataSetDB.DataTableSearchTerms.Rows.Add(DR);
    wb.Navigate(rada_serpURL);
}

private void ClearCookie()
{
    string cPath =
System.Environment.GetFolderPath(Environment.SpecialFolder.Cookies);
    System.IO.DirectoryInfo di = new
System.IO.DirectoryInfo(System.Environment.GetFolderPath(Environment.SpecialFolder
.Cookies));
    foreach (FileInfo file in di.GetFiles("*.txt"))
    {
        file.Delete();
    }
}

```

```

    }

    private void wb_DocumentCompleted(object sender,
    WebBrowserDocumentCompletedEventArgs e)
    {
        tbSbarText.Text = "Горобо";
        byte[] buff = new byte[wb.DocumentStream.Length];
        wb.DocumentStream.Read(buff, 0, buff.Length);
        string lns = System.Text.Encoding.UTF8.GetString(buff).Trim();

        DataRow DR = dataSetDB.DataTableSiteURL.NewRow();
        DR["URL"] = wb.Url.ToString();
        DR["URLDT"] = DateTime.Now.ToString();
        DR["respID"] = 1;
        dataSetDB.DataTableSiteURL.Rows.Add(DR);
        SearchEngineURLGridFill();
    }

    private void wb_NewWindow(object sender, CancelEventArgs e)
    {
        WebBrowser webBrowser = (WebBrowser)sender;
        HTMLElement link = webBrowser.Document.ActiveElement;

        Uri urlNavigated = new Uri(link.GetAttribute("href"));
        string s = urlNavigated.OriginalString;
        Navigate(s);

        e.Cancel = true;
    }

    private void Navigate(string url)
    {
        wb.Navigate(url);
    }

    private void buttonReturn_Click(object sender, EventArgs e)
    {
        URLNavigate(pageCounter);
    }

    private string[] SearchEngineURL(WebBrowser wb, string htmlElementTagName,
    string htmlElementClassName, string htmlElementAttributeName)
    {
        int iLen = wb.Document.GetElementsByTagName(htmlElementTagName).Count;
        string [] EngineURL = new string [iLen];
        int i = 0;
        foreach (HTMLElement HE in
        wb.Document.GetElementsByTagName(htmlElementTagName))
        {
            if (HE.GetAttribute("className").ToString() ==
            htmlElementClassName)
            {
                EngineURL[i] =
                HE.GetAttribute(htmlElementAttributeName).ToString();
                i++;
            }
        }
        return EngineURL;
    }
}

```

```

private void SearchEngineURLGridFill() 242

{
    if (wb.Url.ToString().Contains("http://rada.ua/yandsearch"))
    {
        foreach (string str in SearchEngineURL(wb, "a", "b-serp-
item__title-link", "href"))
        {
            if (str != null && !str.Contains("yabs.rada.ua"))
            {
                DataRow DR = dataSetDB.DataTableSearchEngine.NewRow();
                DR["word"] = tbURL.Text;
                DR["URL"] = str;
                DR["DT"] = DateTime.Now.ToString();
                dataSetDB.DataTableSearchEngine.Rows.Add(DR);
            }
        }
    }
}

private void buttonIUVector_Click(object sender, EventArgs e)
{
    gridControlIUVector.DataSource = null;
    int timeHour = 0;
    int period = 4;
    int vectorCard = 10;

    timeHour = Convert.ToInt16(comboBoxIUHour.Text);
    period = Convert.ToInt16(comboBoxIUPeriod.Text);
    vectorCard = Convert.ToInt16(comboBoxIUCard.Text);

    DataTable _tableTermCounts = new DataTable();
    _tableTermCounts.Columns.Add("Term");
    _tableTermCounts.Columns.Add("Count");

    DataTable tableTermCounts = new DataTable();
    tableTermCounts.Columns.Add("Term");
    tableTermCounts.Columns.Add("Count");
    tableTermCounts.Columns.Add("Wave");

    for (int i = 0; i < (24 / period); i++)
    {
        _tableTermCounts.Clear();
        oneUIVector(timeHour, vectorCard, _tableTermCounts, i, period);
        DataRow[] DR = _tableTermCounts.Select("Term = Term", "Count
DESC").Take(vectorCard).ToArray();

        if (DR.Count() == 0)
            tableTermCounts.Rows.Add("NULL", "NULL", i + 1);
        foreach (DataRow dr in DR)
        {
            tableTermCounts.Rows.Add(dr["Term"].ToString(), dr["Count"].ToString(), i + 1);
        }
    }

    gridControlIUVector.DataSource = tableTermCounts;

    string[] TermWaveDistinct = { "Term", "Wave" };
}

```

```

        DataTable dtDistinctTermsWave = GetDistinctRecords(tableTermCounts,
TermWaveDistinct);
        foreach (DataRow DR in dtDistinctTermsWave.Rows)
        {

            tableTermDinamic.Rows.Add(DR["Term"].ToString(),
DR["Wave"].ToString());
        }

        string[] URLTextDistinct = { "URL" , "pageText" };
        DataTable dtDistinctURLText = GetDistinctRecords(dataSetDB.dtPageText,
URLTextDistinct);

        foreach(DataRow DRTerm in dtDistinctTermsWave.Rows)
        {
            foreach (DataRow DUARL in dtDistinctURLText.Rows)
            {
                DataRow DRPageTextTerm = dataSetDB.dtPageTextTerm.NewRow();
                DRPageTextTerm["URL"] = DUARL["URL"];
                DRPageTextTerm["Term"] = DRTerm["Term"];
                DRPageTextTerm["Count"] =
textTermCount(DUARL["pageText"].ToString(), DRTerm["Term"].ToString());
                DRPageTextTerm["Wave"] = DRTerm["Wave"];
                dataSetDB.dtPageTextTerm.Rows.Add(DRPageTextTerm);
            }
        }

        gridControlIRVector.DataSource = dataSetDB.dtPageTextTerm;

    }

    private void oneUIVector(int timeHour, int vectorCard, DataTable
_tableTermCounts, int i, int period)
    {
        DataRow[] DR = dataSetDB.DataTableSearchTerms.Select("DTHour >= " +
(timeHour + period * (i)).ToString() + " AND DTHour < " + (timeHour + period * (i
+ 1)).ToString());
        DataTable table = new DataTable();
        table.Columns.Add("Term");
        table.Columns.Add("TermDT");
        table.Columns.Add("respID");
        table.Columns.Add("DTHour");
        DR.CopyToDataTable(table, LoadOption.OverwriteChanges);

        var result = from row in table.AsEnumerable()
                    group row by row.Field<string>("Term") into grp
                    select new
                    {
                        Term = grp.Key,
                        MemberCount = grp.Count()
                    };
        foreach (var t in result)
        {
            _tableTermCounts.Rows.Add(t.Term, t.MemberCount);
        }
    }

    private void oneUIVector2(int timeHour, int vectorCard, DataTable
_tableTermCounts, int i, int period)
    {

```

```

        DataRow[] DR = dataSetDB.DataTableSearchTermsNewUser.Select("DTHour >=
" + (timeHour + period * (i)).ToString() + " AND DTHour < " + (timeHour + period *
(i + 1)).ToString());
        DataTable table = new DataTable();
        table.Columns.Add("Term");
        table.Columns.Add("TermDT");
        table.Columns.Add("respID");

        table.Columns.Add("DTHour");
        DR.CopyToDataTable(table, LoadOption.OverwriteChanges);

        var result = from row in table.AsEnumerable()
                    group row by row.Field<string>("Term") into grp
                    select new
                    {
                        Term = grp.Key,
                        MemberCount = grp.Count()
                    };
        foreach (var t in result)
        {
            _tableTermCounts.Rows.Add(t.Term, t.MemberCount);
        }
    }

    private void buttonLoadVisitFile_Click(object sender, EventArgs e)
    {
        if (DialogResult.OK == openFileDialog.ShowDialog())
        {
            openFileDialog.Filter = "Text Files | *.txt";
            openFileDialog.DefaultExt = "txt";
            string[] textData =
System.IO.File.ReadAllLines(openFileDialog.FileName);
            string[] headers = textData[0].Split('\t');
            DataTable dataTable1 = new DataTable();
            foreach (string header in headers)
                dataTable1.Columns.Add(header, typeof(string), null);
            for (int i = 1; i < textData.Length; i++)

dataSetDB.DataTableSearchTerms.Rows.Add(textData[i].Split('\t'));
        }
    }

    private void buttonSaveVisitFile_Click(object sender, EventArgs e)
    {
        if (DialogResult.OK == saveFileDialog.ShowDialog())
        {
            saveFileDialog.Filter = "Text Files | *.txt";
            saveFileDialog.DefaultExt = "txt";
            StreamWriter w = new StreamWriter(saveFileDialog.FileName, false,
Encoding.Default);
            w.WriteLine("Term" + "\t" + "TermDT" + "\t" + "respID" + "\t" +
"DTHour");
            foreach (DataRow row in dataSetDB.DataTableSearchTerms.Rows)
            {
                w.WriteLine(row[0] + "\t" + row[2] + "\t" + row[2] + "\t" +
row[3]);
            }
            w.Close();
        }
    }
}

```

```

private void buttonNextPageIR2_Click(object sender, EventArgs e)
{
    page_move();
}

private void page_move()
{
    if (dtPagesBindingSource.Position != dataSetDB.dtPages.Rows.Count - 1)
    {

        //MessageBox.Show("dtydgsys");
        dtPagesBindingSource.MoveNext();
        URL_listBox.SetSelected(dtPagesBindingSource.Position, true);
        DataRowView DRV = (DataRowView)dtPagesBindingSource.Current;
        page_id_Label.Text = DRV.Row["Page_id"].ToString();
        //MessageBox.Show(page_id_Label.Text);
        page_id = Convert.ToInt32(page_id_Label.Text);
        URLtextBox2.Text = URL_listBox.SelectedItem.ToString();
        wb2.Navigate(URLtextBox2.Text);
        headparaTB.Clear();
        treeTB.Clear();
        linksTB.Clear();
        DOMbutton_Click(null, null);
    }
}

private void buttonNavigate2_Click(object sender, EventArgs e)
{
    wb2.Navigate(URLtextBox2.Text);

    string linkName = link.Name;
    string filterExpression = "[HTML_element_name] = '<' + link.Name +
">' + " AND " + "[take] = 'true'";
    DataRow[] row = dataSetDB.dtTags.Select(filterExpression);
    if (row.Length > 0)
    {
        int HTML_elementID =
Convert.ToInt32(row[0]["HTML_element_id"].ToString());
        switch (linkName)
        {
            case "a":
                if (link.Attributes["href"] != null)
                {

                    string str3 =
link.Attributes["href"].Value.ToString();
                    if (str.Length > 0)
                        TB_filler(linksTB, str3);
                    DataRow DR = dataSetDB.dtHTMLvalue.NewRow();
                    DR["HTML_element_id"] = HTML_elementID;
                    DR["Page_id"] = page_id;
                    DR["HTML_element_value"] = str3;
                    dataSetDB.dtHTMLvalue.Columns[0].AutoIncrement =
true;
                    dataSetDB.dtHTMLvalue.Rows.Add(DR);
                }
                break;
            default:

```



```

        {
            str2 = link.Name + "|" + link.WriteContentTo();
            DataRow DR = dataSetDB.dtHTMLvalue.NewRow();
            DR["HTML_element_id"] = HTML_elementID;
            DR["Page_id"] = page_id;
            DR["HTML_element_value"] = str2;
            dataSetDB.dtHTMLvalue.Rows.Add(DR);
        }
        break;
    }
}

if (str2.ToString() != "")
    TB_filler(headparaTB, str2);
}
}

private void TB_filler(TextBox TB, string str)
{
    TB.AppendText("-" + str + Environment.NewLine + ";");
}

//Одно меню для нескольких textBox - Обработчик.
private void SavefileToolStripMenuItem_Click(object sender, EventArgs e)
{
    var menuItem = (ToolStripMenuItem)sender;
    var toolStrip = menuItem.Owner;
    var ctrl =
(ContextMenuStrip)toolStrip.DataBindings.BindableComponent.DataBindings.Control;
    var textBox = (TextBox)ctrl.SourceControl;
    Save(textBox);
}
private void wb2_DocumentCompleted(object sender,
WebBrowserDocumentCompletedEventArgs e)
{
    tbSbarText2.Text = "Готово";
    URLtextBox2.Text = wb2.Url.ToString();
    dtPageTextFill();
}

private void Save(TextBox TB)
{
    SaveFileDialog saveFileDialog = new SaveFileDialog();
    saveFileDialog.DefaultExt = ".txt";
    saveFileDialog.Filter = "Text files (*.txt)|*.txt|All files (*.*)|*.*";
    if (saveFileDialog.ShowDialog() == DialogResult.OK)
    { 247

        StreamWriter w = new StreamWriter(saveFileDialog.FileName, false,
Encoding.Default);
        foreach (string str in TB.Lines)
        {
            w.WriteLine(str);
        }
        w.Close();
    }
}

private void buttonLoadURL_Click(object sender, EventArgs e)
{

```

```

if (DialogResult.OK == openFileDialog.ShowDialog())
{
    dataSetDB.dtPages.Clear();
    StreamReader r = new StreamReader(openFileDialog.FileName,
Encoding.Default);
    string line;
    while ((line = r.ReadLine()) != null)
    {
        string[] vals = line.Split('\t');

        if (vals.Length >= 3)
        {
            DataRow row = dataSetDB.dtPages.NewRow();
            row["Page_id"] = vals[0];
            row["URL"] = vals[2];
            row["cnt"] = vals[2];
            dataSetDB.dtPages.Rows.Add(row);
            URL_listBox.Items.Add(vals[2]);
        }
    }
    r.Close();
}
DataRowView DRV = (DataRowView)dtPagesBindingSource.Current;
page_id_Label.Text = DRV.Row["Page_id"].ToString();
page_id = Convert.ToInt32(page_id_Label.Text);
URL_listBox.SelectedIndex = 0;
URLtextBox2.Text = URL_listBox.SelectedItem.ToString();
tbSbarText2.Text = "Открытие файла URL";
wb2.Navigate(URLtextBox2.Text);
}

private void buttonLoadTAG_Click(object sender, EventArgs e)
{
    if (DialogResult.OK == openFileDialog.ShowDialog())
    {
        dataSetDB.dtTags.Clear();
        StreamReader r = new StreamReader(openFileDialog.FileName,
Encoding.Default);
        Fname4Save = openFileDialog.FileName;
        string line;
        while ((line = r.ReadLine()) != null)
        {
            string[] vals = line.Split('\t');
            if (vals.Length >= 4)
            {
                DataRow row = dataSetDB.dtTags.NewRow();
                row["HTML_element_id"] = vals[0];
                row["HTML_element_name"] = vals[2];
                row["HTML_element_type"] = vals[2];
                row["take"] = vals[3];

                dataSetDB.dtTags.Rows.Add(row);
                Tag_checkedListBox.Items.Add(vals[2].ToString(),
Convert.ToBoolean(vals[3]));
            }
        }
        r.Close();
    }

    tbSbarText2.Text = "файл TAG-ів";
}

```

```

private void timer_next_URL_Tick(object sender, EventArgs e)
{
    if (dtPagesBindingSource.Position != dataSetDB.dtPages.Rows.Count - 1)
    {
        dtPagesBindingSource.MoveNext();
        URL_listBox.SetSelected(dtPagesBindingSource.Position, true);
        DataRowView DRV = (DataRowView)dtPagesBindingSource.Current;
        page_id_Label.Text = DRV.Row["Page_id"].ToString();
        page_id = Convert.ToInt32(page_id_Label.Text);
        URLtextBox2.Text = URL_listBox.SelectedItem.ToString();
        wb2.Navigate(URLtextBox2.Text);
    }
}

private void buttonLoadURLVisit_Click(object sender, EventArgs e)
{
    dataSetDB.dtPages.Rows.Clear();

    foreach (DataRow DR in dataSetDB.DataTableSiteURL.Rows)
    {
        URL_listBox.Items.Add(DR["URL"].ToString());
        dataSetDB.dtPages.Rows.Add(DR["page_id"], DR["URL"], 1);
    }
    DataRowView DRV = (DataRowView)dtPagesBindingSource.Current;
    page_id_Label.Text = DRV.Row["Page_id"].ToString();
    page_id = Convert.ToInt32(page_id_Label.Text);
    URL_listBox.SelectedIndex = 0;
    URLtextBox2.Text = URL_listBox.SelectedItem.ToString();
    tbSbarText2.Text = "Открытие файла URL";
    wb2.Navigate(URLtextBox2.Text);
}

private void buttonLoadURLSearchEngine_Click(object sender, EventArgs e)
{
    dataSetDB.dtPages.Rows.Clear();

    foreach (DataRow DR in dataSetDB.DataTableSearchEngine.Rows)
    {
        URL_listBox.Items.Add(DR["URL"].ToString());
        dataSetDB.dtPages.Rows.Add(DR["RowID"], DR["URL"], 1);
    }
    DataRowView DRV = (DataRowView)dtPagesBindingSource.Current;
    page_id_Label.Text = DRV.Row["page_id"].ToString();
    page_id = Convert.ToInt32(page_id_Label.Text);
    URL_listBox.SelectedIndex = 0;
    URLtextBox2.Text = URL_listBox.SelectedItem.ToString();
    tbSbarText2.Text = "Открытие файла URL";
    wb2.Navigate(URLtextBox2.Text);
}

private void URL_listBox_SelectedIndexChanged(object sender, EventArgs e)
private void buttonINewUserVector_Click(object sender, EventArgs e)
{
    DataTableSearchTermsNewUserFill();

    gridControlINewUVector.DataSource = null;
    int timeHour = 0;
    int period = 4;
    int vectorCard = 10;
}

```

```

timeHour = Convert.ToInt16(comboBoxIUHour.Text);
period = Convert.ToInt16(comboBoxIUPeriod.Text);
vectorCard = Convert.ToInt16(comboBoxIUCard.Text);

DataTable _tableTermCounts = new DataTable();
_tableTermCounts.Columns.Add("Term");
_tableTermCounts.Columns.Add("Count");

DataTable tableTermCounts = new DataTable();
tableTermCounts.Columns.Add("Term");
tableTermCounts.Columns.Add("Count");
tableTermCounts.Columns.Add("Wave");

for (int i = 0; i < (24 / period); i++)
{
    oneUIVector2(timeHour, vectorCard, _tableTermCounts, i, period);
    DataRow[] DR = _tableTermCounts.Select("Term = Term", "Count
DESC").Take(vectorCard).ToArray();

    if (DR.Count() == 0)
        tableTermCounts.Rows.Add("NULL", "NULL", i + 1);
    foreach (DataRow dr in DR)
    {
        tableTermCounts.Rows.Add(dr["Term"].ToString(),
dr["Count"].ToString(), i + 1);
        tableTermDinamic.Rows.Add(dr["Term"].ToString(), i + 1);
    }

    gridControlINewUVector.DataSource = tableTermCounts;
}

}
private void DataTableSearchTermsNewUserFill ()
{
    if (DialogResult.OK == openFileDialog.ShowDialog())
    {
        dataSetDB.DataTableSearchTermsNewUser.Clear();
        StreamReader r = new StreamReader(openFileDialog.FileName,
Encoding.Default);
        string line;
        while ((line = r.ReadLine()) != null)
        {
            string[] vals = line.Split('\t');

            if (vals.Length >= 4)
            {
                DataRow row =
dataSetDB.DataTableSearchTermsNewUser.NewRow();
                row["Term"] = vals[0]; 251

                row["TermDT"] = vals[2];
                row["respID"] = vals[2];
                row["DTHour"] = vals[3];

                dataSetDB.DataTableSearchTermsNewUser.Rows.Add(row);
            }
        }
        r.Close();
    }
}
}

```

```

private string [] splitTexttoWords(string text)
{
    string[] split = text.Split(new Char[] { ' ', ',', '!', ':', '\t' });

    string[] w = new string[split.Length] ;
    int i = 0;
    foreach (string s in split)
    {

        if (s.Trim().Length >= 5)
        {
            w[i] = s.Trim();
            i++;
        }
    }
    return w;
}

private void buttonINewResourceVector_Click(object sender, EventArgs e)
{
    DataTable DTURLWords = new DataTable();
    DTURLWords.Columns.Add("URL");
    DTURLWords.Columns.Add("Term");

    DataTable DTURLWordsCount = new DataTable();
    DTURLWordsCount.Columns.Add("URL");
    DTURLWordsCount.Columns.Add("Term");
    DTURLWordsCount.Columns.Add("Count");

    string[] URLTextDistinct = { "URL" , "pageText" };
    DataTable dtDistinctURLText = GetDistinctRecords(dataSetDB.dtPageText,
URLTextDistinct);

    foreach (DataRow DUARL in dtDistinctURLText.Rows)
    {
        string pageText = DUARL["pageText"].ToString();
        foreach(string word in splitTexttoWords(pageText).Distinct())
        {
            DTURLWords.Rows.Add(DUARL["URL"].ToString(),word);
        }

        var result = from row in DTURLWords.AsEnumerable()
                    group row by row.Field<string>("Term") into grp
                    select new
                    {
                        Term = grp.Key,
                        MemberCount = grp.Count()
                    };
        foreach (var t in result)
        {
            DTURLWordsCount.Rows.Add(DUARL["URL"], t.Term, t.MemberCount); 252
        }
    }

    gridControlIRNewVector.DataSource = DTURLWordsCount;
    string[] TermText = { "Term" };

```

```
tableTermStatic = GetDistinctRecords(DTURLWordsCount, TermText);

}

private void tabControl1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (tabControl1.SelectedIndex == 3)
    {
        gridControlDinamic.DataSource = tableTermDinamic;
        gridControlStatic.DataSource = tableTermStatic;
    }
}
}
```

ДОДАТОК Г - КОПІЯ ПУБЛІКАЦІЇ