

Frequency analysis software

The software is written in Microsoft Visual C#. The source code or software is tested but Dumeas can not be hold responsible for any loss or damage due to the use of the source code or software provided by Dumeas. For more information or the software pleas contact Dumeas.

Source code

```
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;
using System.Windows.Forms.DataVisualization.Charting;

namespace FrequencyAnalysis
{
    public partial class Frequency_Analysis : Form
    {
        //declare global variable
        int[] characterscount = new int[]{};
        char[] inputtext = new char[]{};
        char[] characters = new char[] { };
        string[] resultsorted = new string[] { };
        char tempchar;
        int charactersarraylength;
        int characterscountarraylength;
        int resultsortedarraylength;
        int charactersarrayindex;
        int tempvalue;
        int countletters;
        int counterchart;

        public Frequency_Analysis()
        {
            InitializeComponent();
        }

        private void analyse_btn_Click(object sender, EventArgs e)
        {
            //set program to zero
            outputtextbox.Clear();
            chart1.Series.Clear();
            Array.Clear(characterscount, 0, characterscount.Length);
            Array.Clear(characters, 0, characters.Length);
            Array.Clear(resultsorted, 0, resultsorted.Length);
            resultsorted = new string[] { };
            chart1.ChartAreas[0].AxisX.Enabled = AxisEnabled.False;
            int charactersarraylength = 0;
            int characterscountarraylength = 0;
            int resultsortedarraylength = 0;
            int charactersarrayindex = 0;
            int tempvalue = 0;
            int countletters = 0;
            int counterchart = 0;
            //get the text from the inputbox and put it in the char array
```

```

char[] inputtext = inputtextbox.Text.ToCharArray();

//output the length of the inputtext
lengthinputtext_label.Text = inputtext.Length + " characters";

//iterate through inputtext array
for (int i = 0; i < inputtext.Length; i++)
{
    //get value from tempvar
    tempchar = inputtext[i];

    //change tempchar to lowercase
    tempchar = Char.ToLower(tempchar);

    //check is the character is in the array else add it
    if (characters.Contains(tempchar))
    {
        //get the index in the charactersarray
        charactersarrayindex = Array.IndexOf(characters, tempchar);
        //get value from the characterscount array
        tempvalue = characterscount[charactersarrayindex];
        //increas tempvalue by 1
        tempvalue = tempvalue + 1;
        //update the characterscount array
        characterscount[charactersarrayindex] = tempvalue;
    }
    else
    {
        //get the length of the characters array
        charactersarraylength = characters.Length + 1;
        //resize character array
        Array.Resize(ref characters, charactersarraylength);
        //add new character to temp array
        characters[charactersarraylength - 1] = tempchar;
        //get the index in the charactersarray
        charactersarrayindex = Array.IndexOf(characters, tempchar);
        //get the length of the characterscount array
        characterscountarraylength = characterscount.Length + 1;
        //resize charactercount array
        Array.Resize(ref characterscount, characterscountarraylength);
        //add new value to temp array
        characterscount[characterscountarraylength - 1] = 1;
    }
}

//get total length of letters in the array
for (int j = 0; j < inputtext.Length; j++)
{
    if (Char.IsLetter(inputtext[j]))
    {
        countletters = countletters + 1;
    }
}

//go trough characterscount array
for (int k = 0; k < characterscount.Length; k++)
{
    //check if the value belongs to a letter
    if (Char.IsLetter(characters[k]))
    {
        characterscount[k] = 100 * characterscount[k] / countletters;
    }
}

```

```

        // Add series and data.
        String tempstring = characters[k].ToString();
        chart1.Series.Add(tempstring);
        DataPoint dp = new DataPoint(counterchart, characterscount[k]);
        chart1.Series[tempstring].Points.Add(dp);
        dp.Label = tempstring;
        chart1.Series[tempstring].Points[0].AxisLabel = tempstring;

        //set the data to resultsorted array
        String tempstring2 = (characters[k] + " = " + characterscount[k] +
"%      ").ToString();
        //get the length of the resultsorted array
        resultsortedarraylength = resultsorted.Length + 1;
        //resize resultsorted array
        Array.Resize(ref resultsorted, resultsortedarraylength);
        //add new value to resultsorted array
        resultsorted[resultsortedarraylength - 1] = tempstring2;
        counterchart++;
    }

    //sort resultarray and output to the outputtextbox
    Array.Sort(resultsorted);
    for (int l = 0; l < resultsorted.Length; l++)
    {
        outputtextbox.AppendText(resultsorted[l]);
    }

    //set the statistics labels
    String temp0 = (countletters + " letters").ToString();
    amountofletterslabel.Text = temp0;
    String temp1 = (inputtext.Length - countletters + " symbols").ToString();
    amountofsymbolslabel.Text = temp1;
}

private void Frequency_Analysis_Load(object sender, EventArgs e)
{
    // Set palette.
    this.chart1.Palette = ChartColorPalette.Bright;

    // Set title.
    this.chart1.Titles.Add("Result of the frequency analysis");
}

private void frequencyanalysis_grpbox_Enter(object sender, EventArgs e)
{
}

private void chart1_Click(object sender, EventArgs e)
{
}

private void linklabel_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    System.Diagnostics.Process.Start("http://www.dumeas.com/");
}

private void label6_Click(object sender, EventArgs e)

```

```

{
}

private void richTextBox1_TextChanged(object sender, EventArgs e)
{
}
}
}

```

User interface

