

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace RememberEachWord
{
    public partial class WordsForm : Form
    {
        private int wordsCount = 0;
        private WordEditorField[] fields = null;
        private bool edit = false;
        private Size origSize;

        public event WordEditorFieldEventHandler WordFieldChanged;

        public int WordsCount
        {
            get { return wordsCount; }
        }

        public bool Edit
        {
            get
            {
                return edit;
            }
            set
            {
                edit = value;
                foreach(WordEditorField wef in fields)
                {
                    wef.Edit = edit;
                }
            }
        }

        public Font VisFont
        {
            set
            {
                for (int i = 0; i < wordsCount; i++)
                {
                    fields[i].CurrentVisibleFont = value;
                }
            }
        }

        public Font UnvFont
        {
            set
            {
                for (int i = 0; i < wordsCount; i++)
                {
                    fields[i].CurrentUnvisibleFont = value;
                }
            }
        }

        public Color VisColor
        {
            set
            {
                for (int i = 0; i < wordsCount; i++)
                {
                    fields[i].CurrentVisibleColor = value;
                }
            }
        }
    }
}
```

```
public Color UnvColor
{
    set
    {
        for (int i = 0; i < wordsCount; i++)
        {
            fields[i].CurrentUnvisibleColor = value;
        }
    }
}

public WordsForm()
{
    InitializeComponent();

    foreach (Control con in Controls)
    {
        if (con is WordEditorField)
        {
            wordsCount++;
        }
    }

    fields = new WordEditorField[wordsCount];

    int i = wordsCount - 1;
    foreach (Control con in Controls)
    {
        if (con is WordEditorField)
        {
            fields[i] = con as WordEditorField;
            fields[i].ItemIndex = i;
            fields[i].State = false;
            fields[i].FieldChanged += WordsForm_FieldChanged;
            fields[i].ItemDoubleClick += WordsForm_ItemDoubleClick;
            fields[i].Edit = edit;
            i--;
        }
    }
}

void WordsForm_ItemDoubleClick(object sender, WordEditorFieldEventArgs e)
{
    for (int i = 0; i < wordsCount; i++)
    {
        if (i != e.Index) fields[i].EndEdit();
    }
}

void WordsForm_FieldChanged(object sender, WordEditorFieldEventArgs e)
{
    // Индекс поля в событии
    // Передаем событие в главное окно
    if (WordFieldChanged != null) WordFieldChanged(this, e);
}

public void SetItemText(string txt, int index)
{
    if ((index < wordsCount) && (index >= 0))
    {
        fields[index].ItemText = txt;
    }
}

public string GetItemText(int index)
{
    if ((index < wordsCount) && (index >= 0))
    {
        return fields[index].ItemText;
    }
    else
    {

```

```

        return "";
    }
}

public void SetItemState(bool st, int index)
{
    if ((index < wordsCount) && (index >= 0))
    {
        fields[index].State = st;
    }
}

public bool GetItemState(int index)
{
    if ((index < wordsCount) && (index >= 0))
    {
        return fields[index].State;
    }
    else
    {
        return false;
    }
}

public Point GetRelItemLocation(int index)
{
    if ((index < wordsCount) && (index >= 0))
    {
        return fields[index].OrigLocation;
    }
    else
    {
        return new Point(0, 0);
    }
}

public Size GetRelItemSize(int index)
{
    if ((index < wordsCount) && (index >= 0))
    {
        return fields[index].OrigSize;
    }
    else
    {
        return new Size(0, 0);
    }
}

public Size GetPictureSize()
{
    float kX = (float)this.ClientSize.Width / (float)this.BackgroundImage.Width;
    float kY = (float)this.ClientSize.Height / (float)this.BackgroundImage.Height;
    float k;
    if (kX < kY)
    {
        k = kX;
    }
    else
    {
        k = kY;
    }
    int width = MulIntFloat(this.BackgroundImage.Width, k);
    int height = MulIntFloat(this.BackgroundImage.Height, k);
    return new Size(width, height);
}

public Point GetPictureLocation()
{
    Size picSize = GetPictureSize();
    return new Point((this.ClientSize.Width - picSize.Width) / 2, (this.ClientSize.Height - picSize.
Height) / 2);
}

public void StoreOrigParameters()

```

```

    {
        Point pitch = GetPictureLocation();

        origSize = this.GetPictureSize();

        for (int i = 0; i < fields.Length; i++)
        {
            fields[i].StoreOrigParameters(new Point(fields[i].Location.X - pitch.X, fields[i].Location.Y
- pitch.Y));
        }
    }

    protected override void OnResize(EventArgs e)
    {
        base.OnResize(e);

        if (this.Visible)
        {
            if (fields != null)
            {
                if (fields.Length > 0)
                {
                    Size picSize = GetPictureSize();
                    float k = (float)picSize.Width / (float)origSize.Width;

                    Point pitch = GetPictureLocation();

                    for (int i = 0; i < fields.Length; i++)
                    {
                        // Новые координаты
                        fields[i].Location = new Point(MulIntFloat(fields[i].OrigLocation.X, k) + pitch.X
, MulIntFloat(fields[i].OrigLocation.Y, k) + pitch.Y);

                        // Новый размер
                        fields[i].Size = new Size(MulIntFloat(fields[i].OrigSize.Width, k), MulIntFloat
(fields[i].OrigSize.Height, k));

                        // Шрифты
                        fields[i].Zoom = k;
                    }
                }
            }
        }

        public int MulIntFloat(int first, float second)
        {
            return (int)((float)first * second);
        }

        public void ClearForm()
        {
            for (int i = 0; i < wordsCount; i++)
            {
                fields[i].ItemText = "";
                fields[i].State = false;
            }
        }

        //----- Для отладки -----

        //Font fnt = new Font("Jokerman", 24f, FontStyle.Regular);
        //protected override void OnPaint(PaintEventArgs e)
        //{
        //    base.OnPaint(e);

        //    Rectangle rect = new Rectangle(0, 0, this.ClientRectangle.Width, this.ClientRectangle.Height *
2 / 3);
        //    Graphics gr = e.Graphics;
        //    StringFormat sf = new StringFormat();
        //    sf.Alignment = StringAlignment.Center;
        //    sf.LineAlignment = StringAlignment.Center;
        //    gr.DrawString("Test string: Hello, World!", fields[0].CurrentFont, new SolidBrush(Color.Black),
rect, sf);
    }

```

```
        //}  
    }  
}
```