Федеральное агентство связи

Федеральное государственное бюджетное образовательное учреждение

высшее образования

«Сибирский Государственный Университет телекоммуникаций и информатики»

Лабораторная работа №2

Выполнил студент 2 курса

ИВТ, гр ИП-814

Якунин Андрей

Проверил

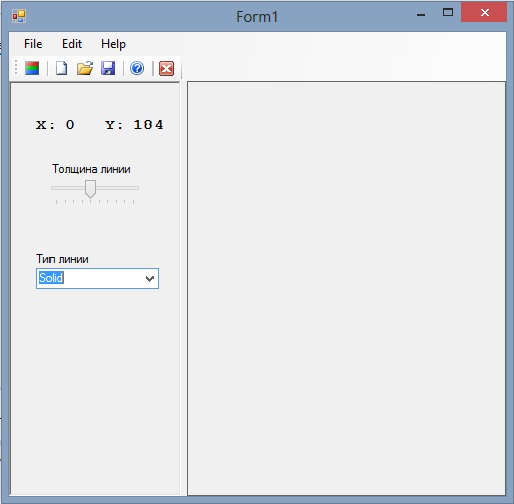
Ассистент ПМиК

Суходоева Наталья Николаевна

Новосибирск 2020

**Задание**

Вид формы после создания визуальной части показан на рисунке:

****

**ComboBox**

**PictureBox**

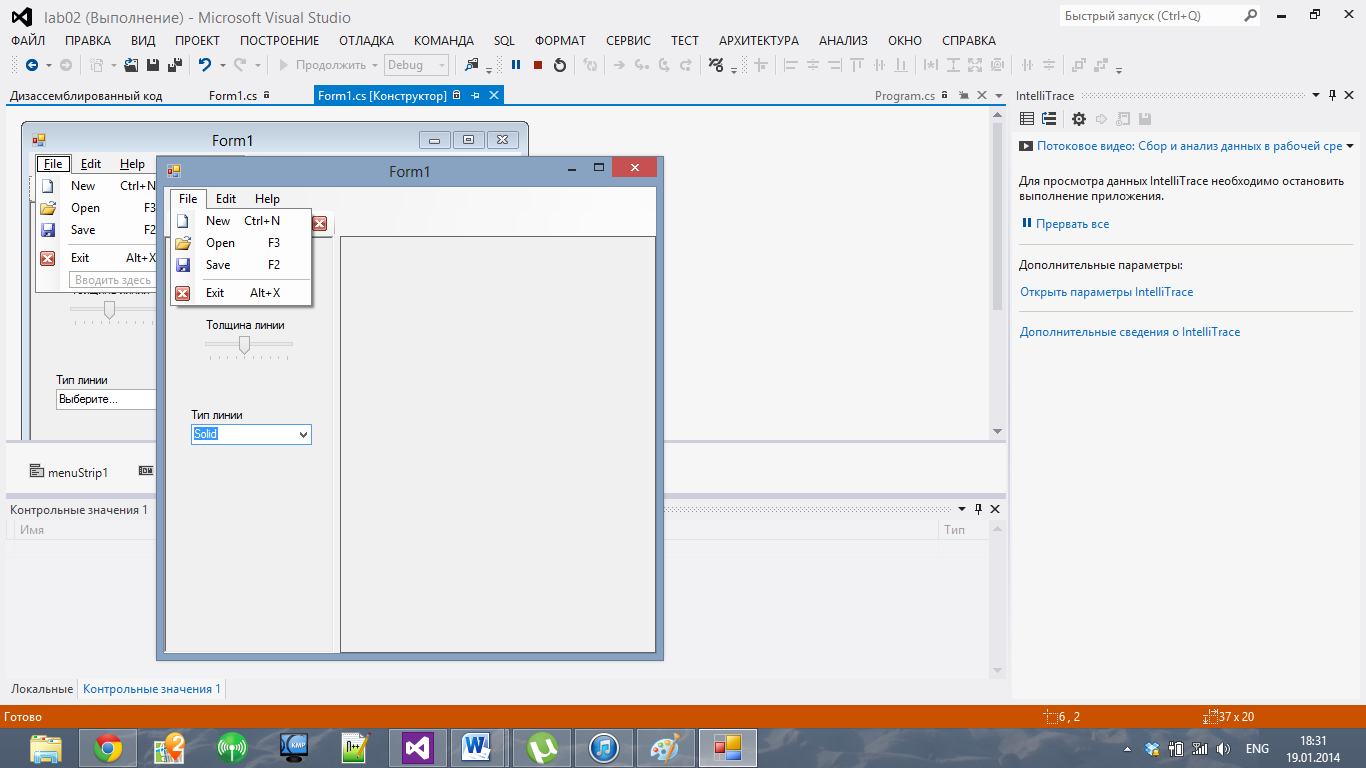
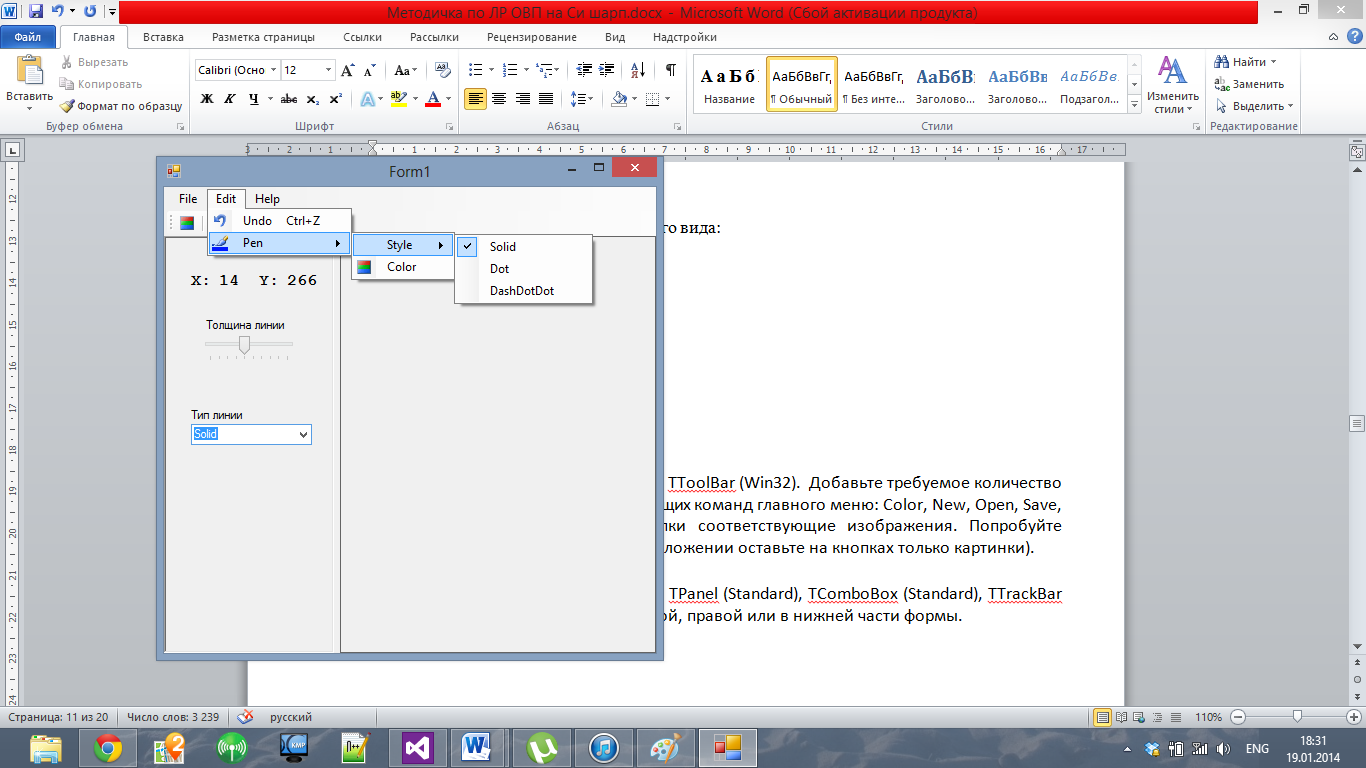
**TrackBar**

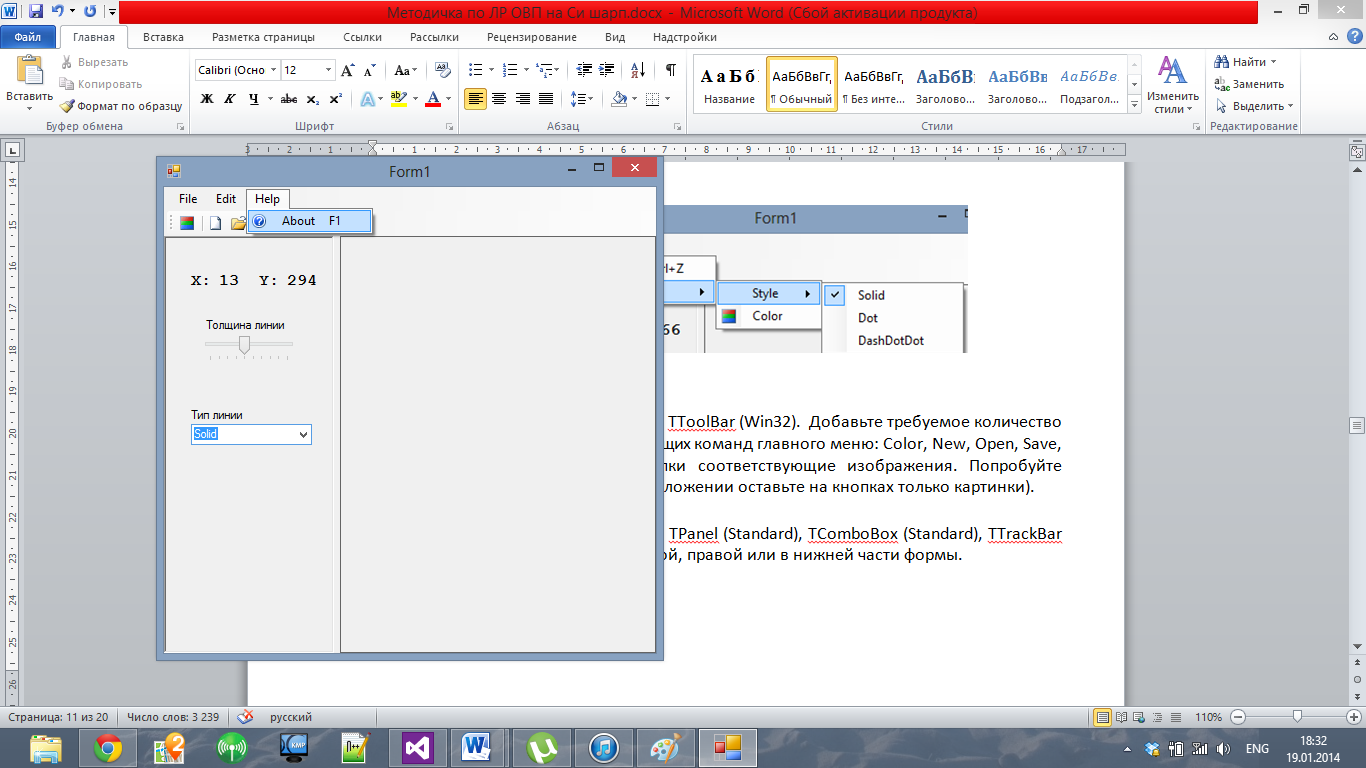
**Panel**

**ToolStrip**

**Создание визуальной части.**

1. Создайте главное меню следующего вида:



1. Поместите на форму компоненты ToolStrip. Добавьте требуемое количество кнопок для дублирования следующих команд главного меню: Color, New, Open, Save, About, Exit. Поместите на кнопки соответствующие изображения. Попробуйте разместить и текст (в готовом Приложении оставьте на кнопках только картинки).
2. Поместите на форму компоненты Panel, ComboBox, TrackBar. Панель разместите в левой, правой или в нижней части формы.
3. Поместите на форму компонент PictureBox. На него поместите четыре компонента label для отображения координат указателя мыши в виде:

65

9

X:

Y:

**Создание невизуальной части.**

1. Запрограммируйте работу всех пунктов меню File. Для этого поместите на форму компоненты OpenFileDialog и SaveFileDialog (см.пример в лабораторной 1).
2. Запрограммируйте работу пункта About меню Help: должна появляться форма с кратким описанием возможностей Вашего графического редактора (см.пример в лабораторной 1).
3. В разделе private опишите переменную Dragging типа boolean, которая принимает значение True, если мышь находиться в режиме рисования. В обработчике события OnCreate формы установите значение этой переменной равным False.
4. В процедуре обработки события OnMouseDown в случае нажатия левой кнопки установите режим рисования (переменная Dragging), установите графический указатель в позицию указателя мыши и отобразите текущие координаты указателя мыши в соответствующих компонентах label
5. В процедуре обработки события OnMouseMove, если установлен режим рисования, проводится прямая линия из текущей позиции графического указателя в новую позицию указателя мыши. После выполнения метода AddLine графический указатель автоматически перемещается в конечную точку. В процедуре OnMouseMove получают новые значения координаты указателя мыши, отображаемые в соответствующих компонентах Label.
6. В процедуру обработки события OnMouseUp добавьте отмену использования мыши для рисования.
7. Запрограммируйте работу пункта меню Undo (стирает рисунок) по двойному щелчку левой кнопки мыши.

*Задание 2.* Создание дополнительных сервисных возможностей

1. Запрограммируйте работу всех пунктов подменю Pen меню Edit. Эти команды устанавливают стиль и цвет рисования. По команде Color должно открываться окно диалога для выбора цвета (компонент ColorDialog закладки Dialogs).

Сделайте выбор толщины линии с помощью компонента TrackBar. У TrackBar установите свойства Minimum=1 Maximum=10 Value =5

\_currentPen.Width = trkPenWidth.Value;

1. Запрограммируйте «стирательную резинку». При нажатии правой кнопки мыши изображение стирается. Величина резинки определяется текущей толщиной линии. Для отображения резинки можно использовать нестандартный курсор.

**Листинг**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Drawing.Imaging;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Laba2

{

public partial class Form1 : Form

{

bool \_dragging = false;

public Color CurrentColor = Color.Black;

Point CurrentPoint;

Point PrevPoint;

Graphics g;

public int red = 0, green = 0, blue = 0;

int choosen\_style = 1;

public Form1()

{

InitializeComponent();

var bitmap = new Bitmap(picDrawingSurface.Width, picDrawingSurface.Height);

picDrawingSurface.BackColor = Color.White;

picDrawingSurface.Image = bitmap;

g = Graphics.FromImage(bitmap);

comboBox1.SelectedIndex = 0;

}

private void choose\_style()

{

choosen\_style = comboBox1.SelectedIndex;

}

private void paint\_simple()

{

Pen p = new Pen(CurrentColor, trackBar1.Value);

if (solidToolStripMenuItem.Checked)

p.DashStyle = DashStyle.Solid;

if (dotToolStripMenuItem.Checked)

p.DashStyle = DashStyle.Dot;

if (dashDotDotToolStripMenuItem.Checked)

p.DashStyle = DashStyle.DashDotDot;

//g.DrawLine(p, new Point(0, 0), new Point(picDrawingSurface.Width, picDrawingSurface.Height)); //работают стили

g.DrawLine(p, PrevPoint, CurrentPoint);

picDrawingSurface.Invalidate();

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void newToolStripMenuItem\_Click(object sender, EventArgs e)

{

//g.Dispose

picDrawingSurface.Image = null;

var bitmap = new Bitmap(picDrawingSurface.Width, picDrawingSurface.Height);

picDrawingSurface.BackColor = Color.White;

picDrawingSurface.Image = bitmap;

g = Graphics.FromImage(bitmap);

}

private void openToolStripMenuItem\_Click(object sender, EventArgs e)

{

OpenFileDialog OP = new OpenFileDialog();

OP.Filter = "JPEG Image|\*.jpg|Bitmap Image|\*.bmp|GIF Image|\*.gif|PNG Image|\*.png";

OP.Title = "Open an Image File";

OP.FilterIndex = 1; //По умолчанию будет выбрано первое расширение \*.jpg

if (OP.ShowDialog() != DialogResult.Cancel)

picDrawingSurface.Load(OP.FileName);

picDrawingSurface.AutoSize = true;

}

private void saveToolStripMenuItem\_Click\_1(object sender, EventArgs e)

{

SaveFileDialog SaveDlg = new SaveFileDialog();

SaveDlg.Filter = "JPEG Image|\*.jpg|Bitmap Image|\*.bmp|GIF Image|\*.gif|PNG Image|\*.png";

SaveDlg.Title = "Save an Image File";

SaveDlg.FilterIndex = 4; //По умолчанию будет выбрано последнее расширение \*.png

SaveDlg.ShowDialog();

if (SaveDlg.FileName != "") //Если введено не пустое имя

{

System.IO.FileStream fs =

(System.IO.FileStream)SaveDlg.OpenFile();

switch (SaveDlg.FilterIndex)

{

case 1:

this.picDrawingSurface.Image.Save(fs, ImageFormat.Jpeg);

break;

case 2:

this.picDrawingSurface.Image.Save(fs, ImageFormat.Bmp);

break;

case 3:

this.picDrawingSurface.Image.Save(fs, ImageFormat.Gif);

break;

case 4:

this.picDrawingSurface.Image.Save(fs, ImageFormat.Png);

break;

}

fs.Close();

}

}

private void toolStripButton2\_Click(object sender, EventArgs e)

{

picDrawingSurface.Image = null;

var bitmap = new Bitmap(picDrawingSurface.Width, picDrawingSurface.Height);

picDrawingSurface.BackColor = Color.White;

picDrawingSurface.Image = bitmap;

g = Graphics.FromImage(bitmap);

}

private void toolStripButton3\_Click(object sender, EventArgs e)

{

OpenFileDialog OP = new OpenFileDialog();

OP.Filter = "JPEG Image|\*.jpg|Bitmap Image|\*.bmp|GIF Image|\*.gif|PNG Image|\*.png";

OP.Title = "Open an Image File";

OP.FilterIndex = 1; //По умолчанию будет выбрано первое расширение \*.jpg

if (OP.ShowDialog() != DialogResult.Cancel)

picDrawingSurface.Load(OP.FileName);

picDrawingSurface.AutoSize = true;

}

private void toolStripButton4\_Click(object sender, EventArgs e)

{

SaveFileDialog SaveDlg = new SaveFileDialog();

SaveDlg.Filter = "JPEG Image|\*.jpg|Bitmap Image|\*.bmp|GIF Image|\*.gif|PNG Image|\*.png";

SaveDlg.Title = "Save an Image File";

SaveDlg.FilterIndex = 4; //По умолчанию будет выбрано последнее расширение \*.png

SaveDlg.ShowDialog();

if (SaveDlg.FileName != "") //Если введено не пустое имя

{

System.IO.FileStream fs =

(System.IO.FileStream)SaveDlg.OpenFile();

switch (SaveDlg.FilterIndex)

{

case 1:

this.picDrawingSurface.Image.Save(fs, ImageFormat.Jpeg);

break;

case 2:

this.picDrawingSurface.Image.Save(fs, ImageFormat.Bmp);

break;

case 3:

this.picDrawingSurface.Image.Save(fs, ImageFormat.Gif);

break;

case 4:

this.picDrawingSurface.Image.Save(fs, ImageFormat.Png);

break;

}

fs.Close();

}

}

private void toolStripButton5\_Click(object sender, EventArgs e)

{

MessageBox.Show("Якунин Андрей и Парфенцов Семен ИП-814");

}

private void aboutToolStripMenuItem\_Click(object sender, EventArgs e)

{

MessageBox.Show("Якунин Андрей и Парфенцов Семен ИП-814");

}

private void pictureBox1\_MouseDown(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

\_dragging = true;

CurrentPoint = e.Location;

//\_oldLocation = e.Location;

//\_currentPath = new GraphicsPath();

}

}

private void picDrawingSurface\_MouseUp(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

\_dragging = false;

//picDrawingSurface.Update();

try

{

//\_currentPath.Dispose();

}

catch

{

};

}

}

private void picDrawingSurface\_MouseMove(object sender, MouseEventArgs e)

{

lblX.Text = e.X.ToString();

lblY.Text = e.Y.ToString();

if (\_dragging)

{

PrevPoint = CurrentPoint;

CurrentPoint = e.Location;

//pen

paint\_simple();

//Graphics g = Graphics.FromImage(picDrawingSurface.Image);

// \_currentPath.AddLine(\_oldLocation, e.Location);

//g.DrawPath(\_currentPen, \_currentPath);

// \_oldLocation = e.Location;

//g.Dispose();

//picDrawingSurface.Invalidate();

}

}

private void undoToolStripMenuItem\_DoubleClick(object sender, EventArgs e)

{

Graphics g = Graphics.FromImage(picDrawingSurface.Image);

//g.Clear(\_bgColor);

g.Dispose();

picDrawingSurface.Invalidate();

}

private void toolStripButton6\_Click(object sender, EventArgs e)

{

this.Close();

}

private void exitToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Close();

}

private void colorToolStripMenuItem\_Click(object sender, EventArgs e)

{

ColorDialog dlg = new ColorDialog();

if (dlg.ShowDialog() == DialogResult.OK)

{

CurrentColor = dlg.Color;

}

}

private void solidToolStripMenuItem\_Click(object sender, EventArgs e)

{

solidToolStripMenuItem.Checked = true;

dotToolStripMenuItem.Checked = false;

dashDotDotToolStripMenuItem.Checked = false;

comboBox1.SelectedIndex = 0;

}

private void dotToolStripMenuItem\_Click(object sender, EventArgs e)

{

dotToolStripMenuItem.Checked = true;

solidToolStripMenuItem.Checked = false;

dashDotDotToolStripMenuItem.Checked = false;

comboBox1.SelectedIndex = 1;

}

private void dashDotDotToolStripMenuItem\_Click(object sender, EventArgs e)

{

dashDotDotToolStripMenuItem.Checked = true;

dotToolStripMenuItem.Checked = false;

solidToolStripMenuItem.Checked = false;

comboBox1.SelectedIndex = 2;

}

private void toolStripButton1\_Click(object sender, EventArgs e)

{

Form2 dlg = new Form2();

dlg.Owner = this;

dlg.ShowDialog();

}

public void myColorToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form2 dlg = new Form2();

dlg.Owner = this;

dlg.ShowDialog();

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

if (comboBox1.SelectedIndex == 0)

{

solidToolStripMenuItem.Checked = true;

dotToolStripMenuItem.Checked = false;

dashDotDotToolStripMenuItem.Checked = false;

}

if (comboBox1.SelectedIndex == 1)

{

dotToolStripMenuItem.Checked = true;

solidToolStripMenuItem.Checked = false;

dashDotDotToolStripMenuItem.Checked = false;

}

if (comboBox1.SelectedIndex == 2)

{

dashDotDotToolStripMenuItem.Checked = true;

dotToolStripMenuItem.Checked = false;

solidToolStripMenuItem.Checked = false;

}

}

bool clear\_pen = false;

private void erasebtn\_Click(object sender, EventArgs e)

{

clear\_pen = !clear\_pen;

if (clear\_pen)

{

CurrentColor = Color.White;

checkBox\_erase.Checked = true;

}

else

{

CurrentColor = Color.Black;

checkBox\_erase.Checked = false;

}

}

private void picDrawingSurface\_DoubleClick\_1(object sender, EventArgs e)

{

}

private void helpToolStripMenuItem\_Click(object sender, EventArgs e)

{

}

private void undoToolStripMenuItem\_Click(object sender, EventArgs e)

{

Graphics g = Graphics.FromImage(picDrawingSurface.Image);

g.Clear(Color.White);

g.Dispose();

picDrawingSurface.Invalidate();

}

private void trackBar1\_Scroll(object sender, EventArgs e)

{

}

private void penToolStripMenuItem\_Click(object sender, EventArgs e)

{

}

private void lblX\_Click(object sender, EventArgs e)

{

}

private void picDrawingSurface\_Click(object sender, EventArgs e)

{

}

}

}

**Скриншоты**

