

2 Инструкция по работе с программой

2.1 Запуск программы

Запустить файл MDI.exe. Откроется главное окно программы в соответствии с рисунком 1.

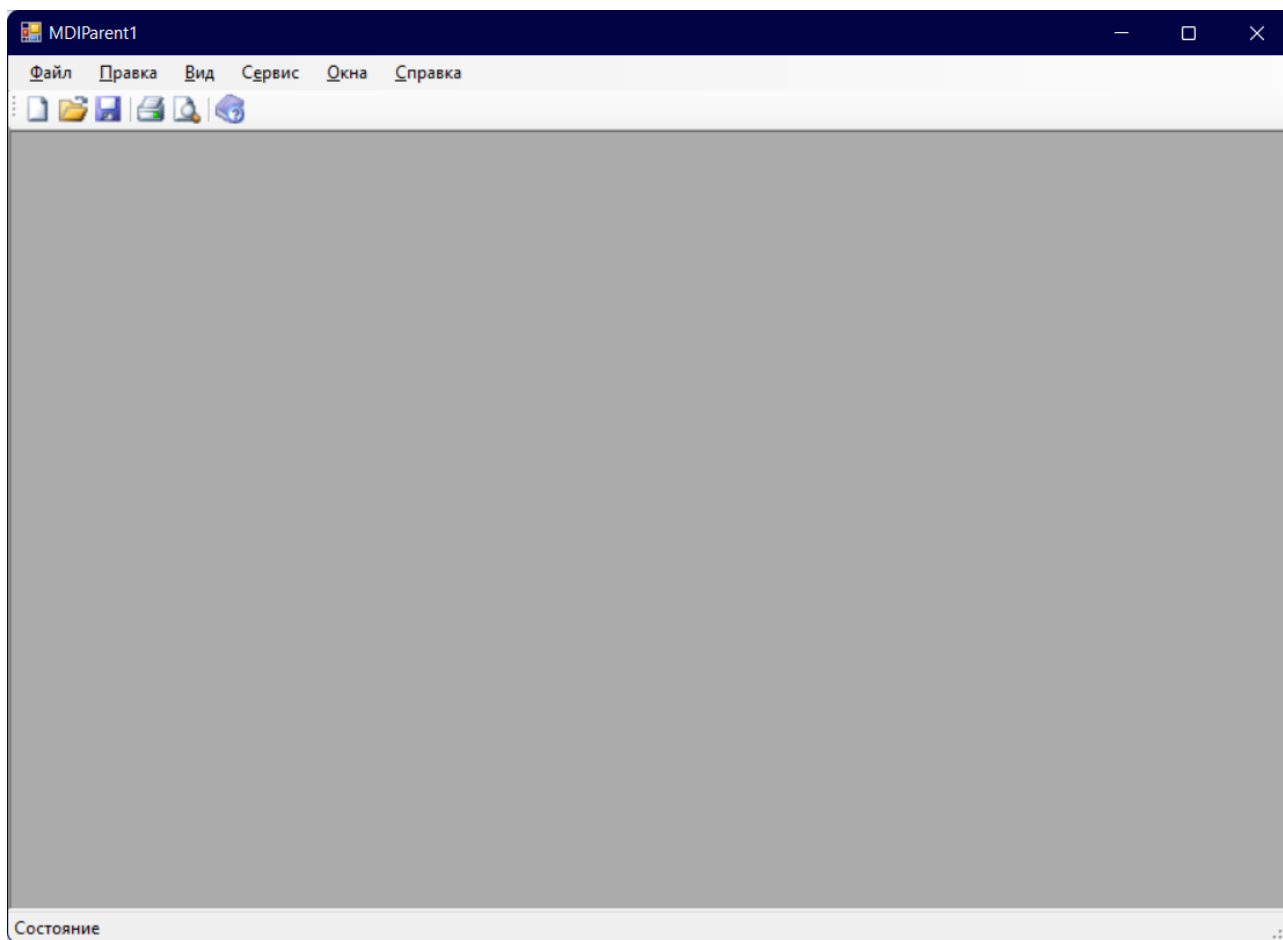


Рисунок 1 – Главное окно программы

2.2 Элементы управления

Управление функционалом программы осуществляется через элементы управления.

2.2.1 Главное меню

Элементы управления главного меню скомпонованы по группам и выполняют следующие функции:

а) Файл

- 1) Создать – создает новое окно в соответствии с рисунком 2

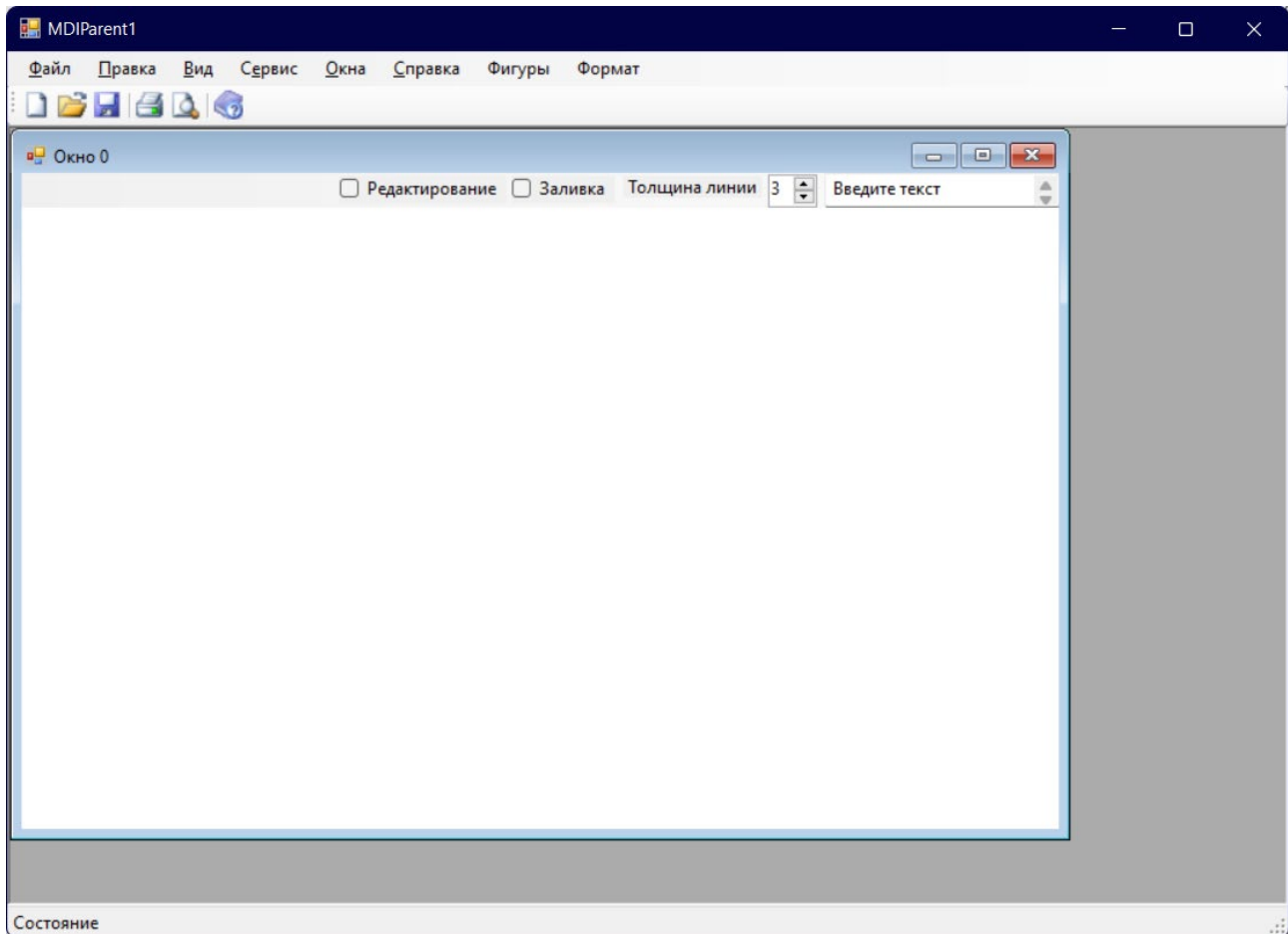


Рисунок 2 – Создание нового окна

- 2) Открыть – вызывает файловый диалог в соответствии с рисунком 3 для выбора файла, при отсутствии открытых окон дополнительно открывает новое окно согласно рисунка 4

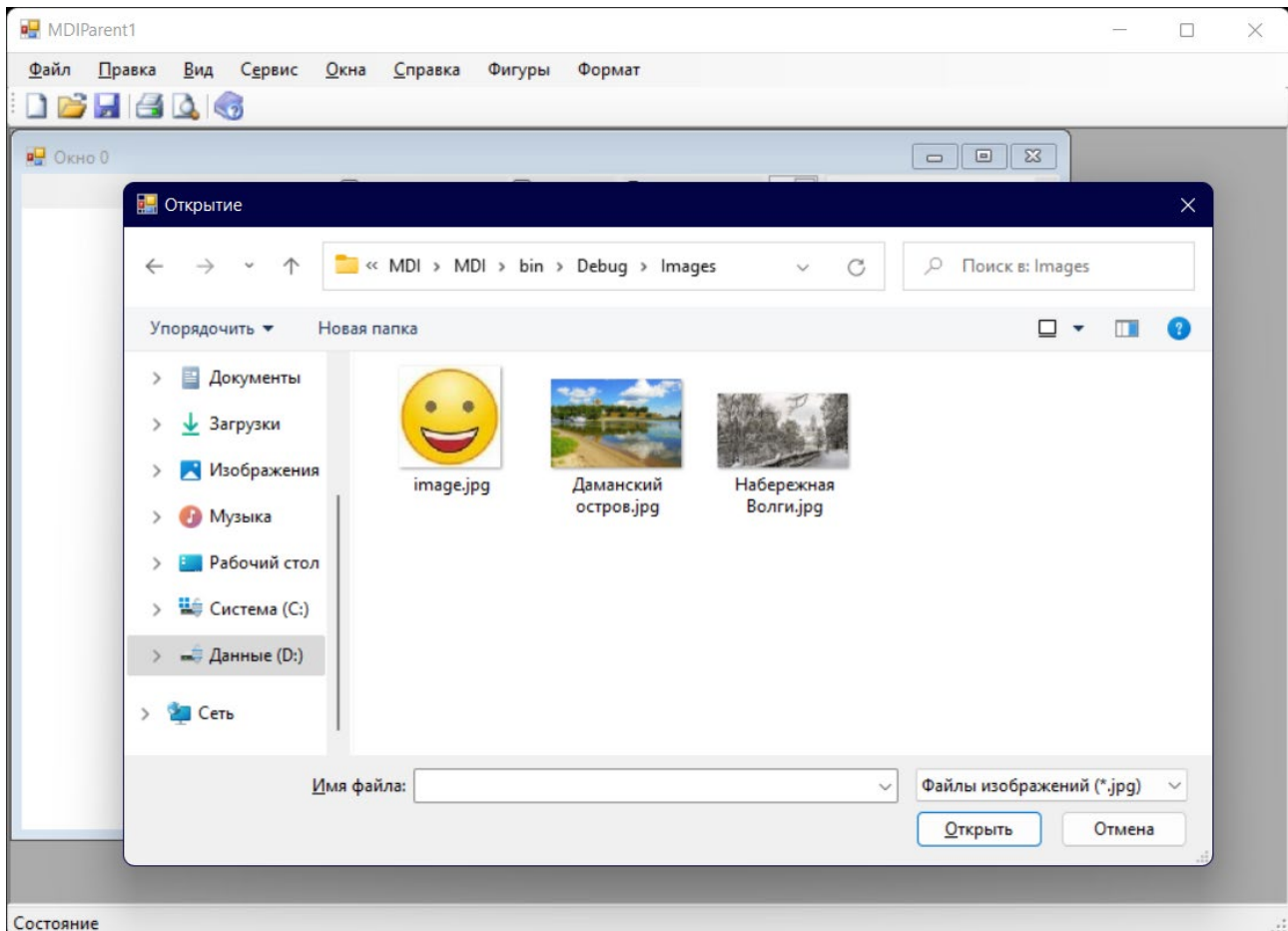


Рисунок 3 – Файловый диалог

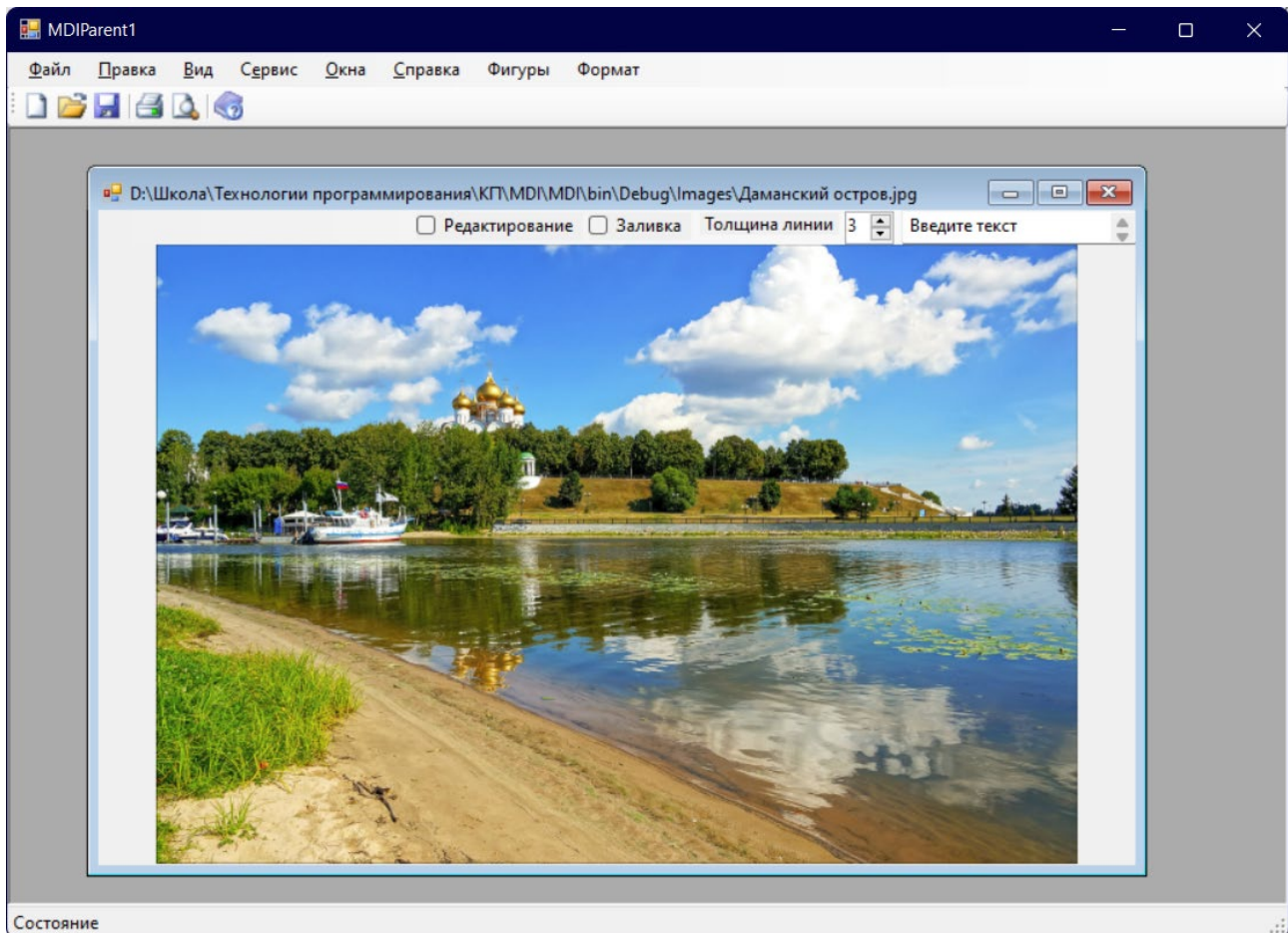


Рисунок 4 – Открытие файла в новом окне

3) Сохранить – сохраняет изменения, внесенные в ранее открытый файл, при редактировании в новом окне открывает файловый диалог

4) Сохранить как – открывает файловый диалог для сохранения файла

5) Выход – закрывает программу

б) Правка

1) Очистить – очищает область редактирования окна, текстовое поле и возвращает окно в режим просмотра

2) Вырезать – вырезает выделенный фрагмент согласно рисунка 5

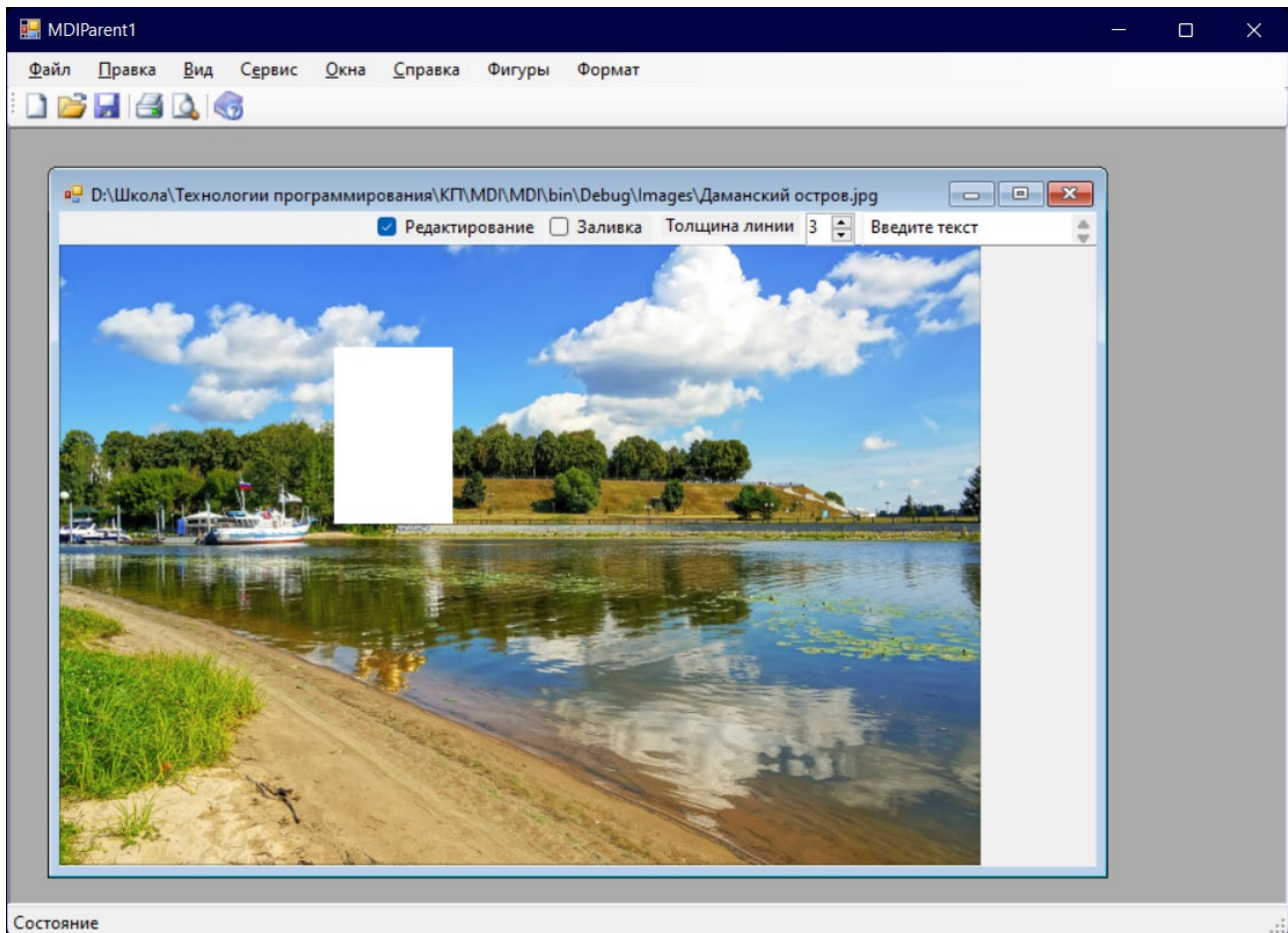


Рисунок 5 – вырезание фрагмента

3) Копировать – копирует выделенный фрагмент

4) Вставить – вставляет выделенный фрагмент в указанное место
активного окна согласно рисунка 6

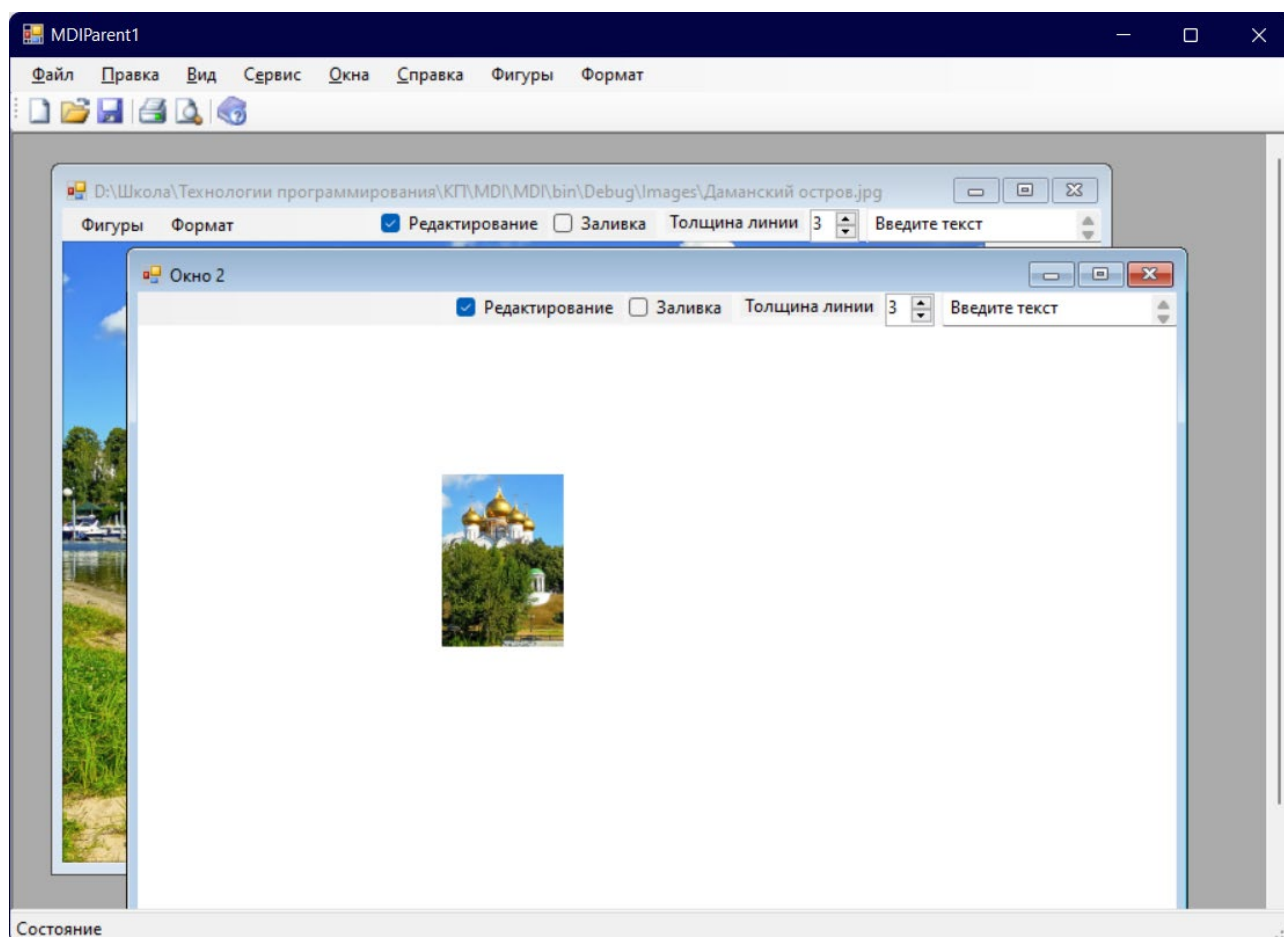


Рисунок 6 – Вставка выделенного фрагмента

5) Выделить – выделяет фрагмент изображения

6) Выделить все – выделяет всю область редактирования активного

окна

в) Вид

1) Панель инструментов – отображает или скрывает панель инструментов

2) Строка состояния – отображает или скрывает строку состояния

г) Окна

1) Новое окно – дублирует функции п. а.1

2) Каскадом – располагает открытые окна каскадом

3) Слева направо – располагает открытые окна слева направо

4) Сверху вниз – располагает открытые окна сверху вниз

5) Закрыть все – закрывает все окна

д) Фигуры (управляет функциями активного окна, элемент скрыт при отсутствии активных окон)

1) Линия – рисует прямую линию согласно рисунка 7

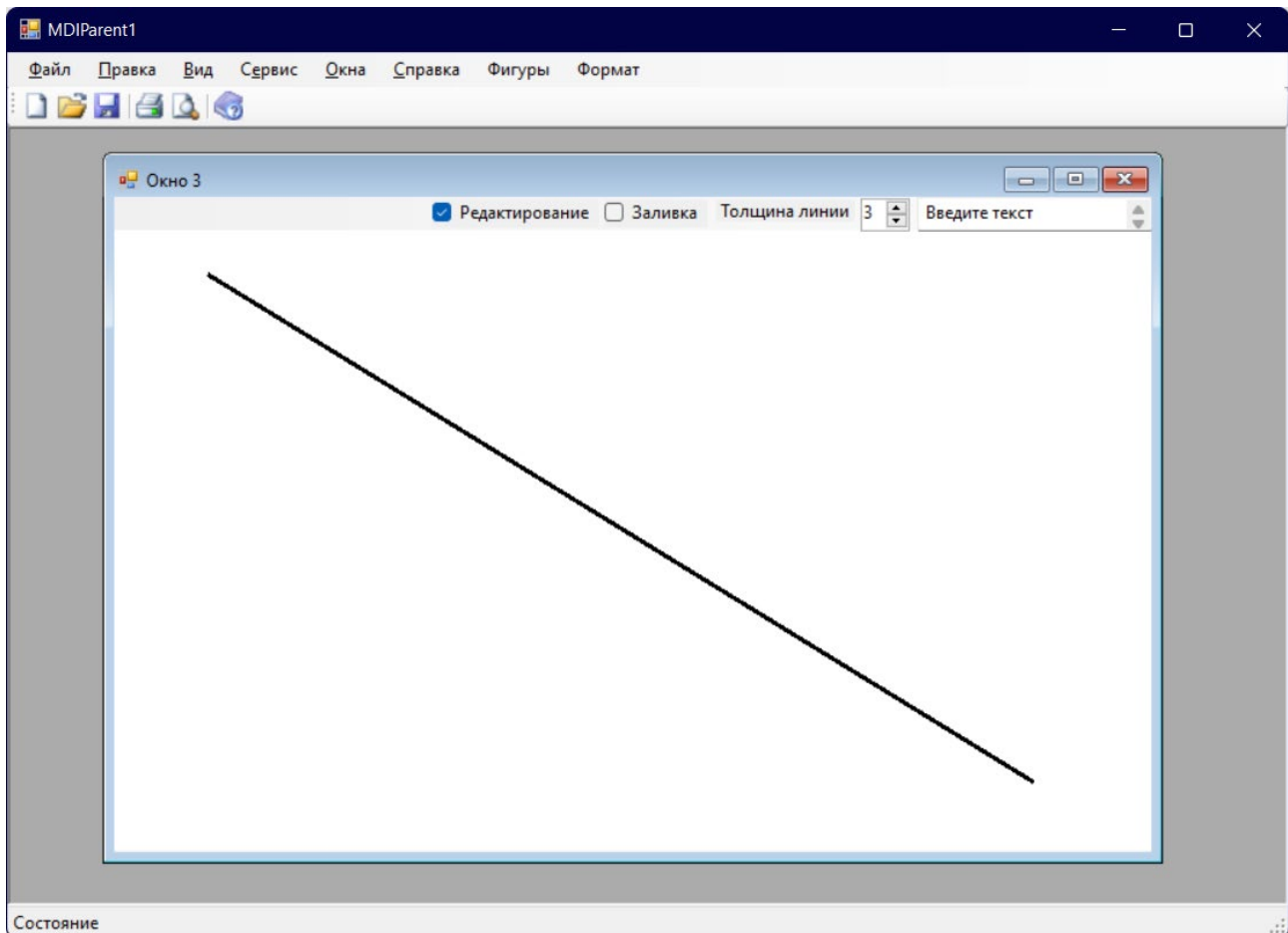


Рисунок 7 – Рисование прямой линии

- 2) Кривая – рисует кривую линия
- 3) Треугольник – рисует треугольник
- 4) Прямоугольник – рисует прямоугольник
- 5) Шестиугольник – рисует шестиугольник
- 6) Эллипс – рисует эллипс
- 7) Текст – размещает в указанном месте поля формы текст из тестовой панели
- 8) Изображение – вставляет в указанное место поля формы изображение небольших размеров
- 9) Ластик – удаляет часть изображения перетаскиванием курсора мыши согласно рисунка 8

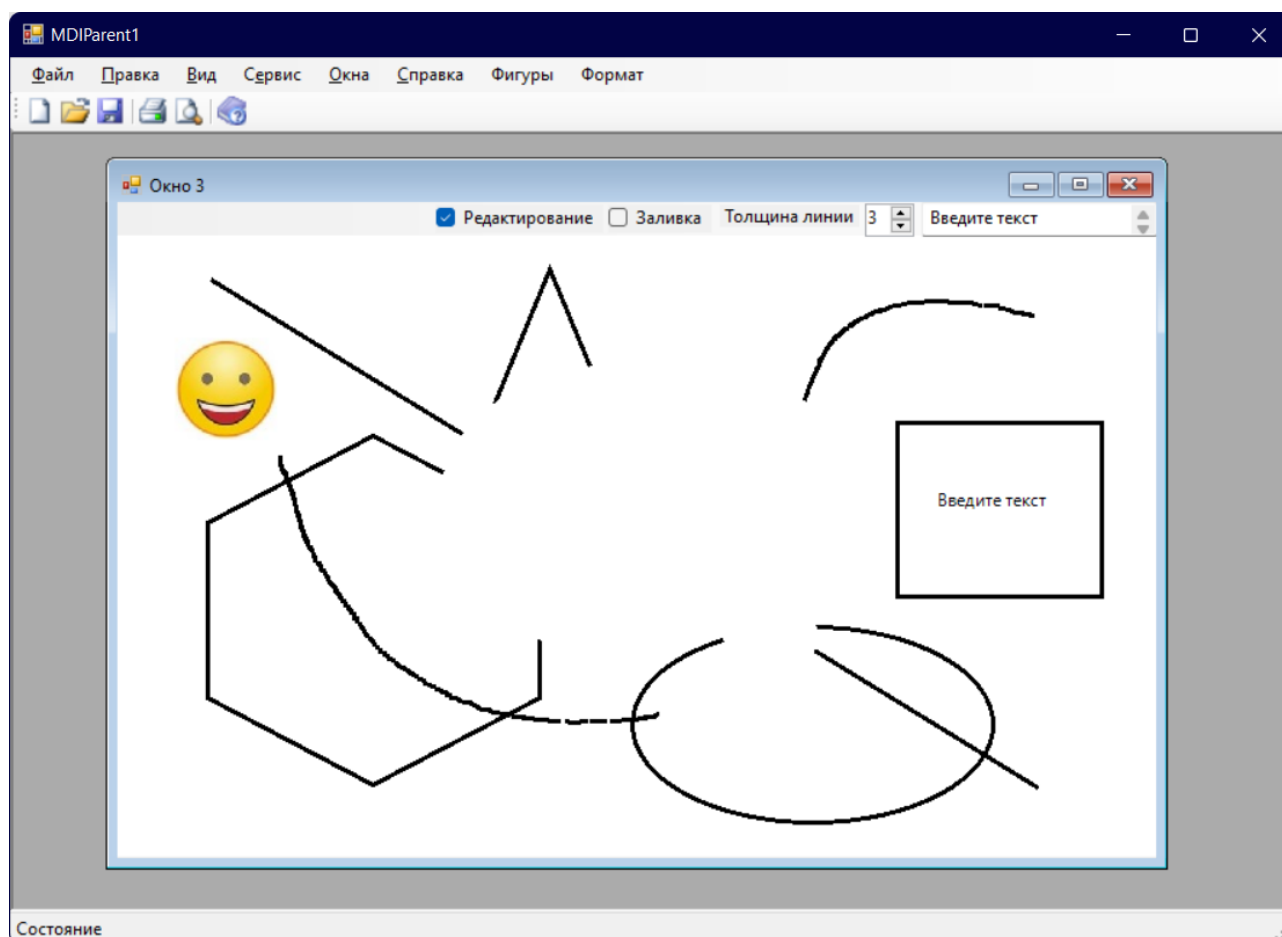


Рисунок 8 – Удаление части изображения инструментом Ластик

е) Формат (управляет функциями активного окна и скрыт при отсутствии активных окон)

1) Цвет линии – устанавливает выбранный в диалоге выбора цвета цвет линии согласно рисунка 9

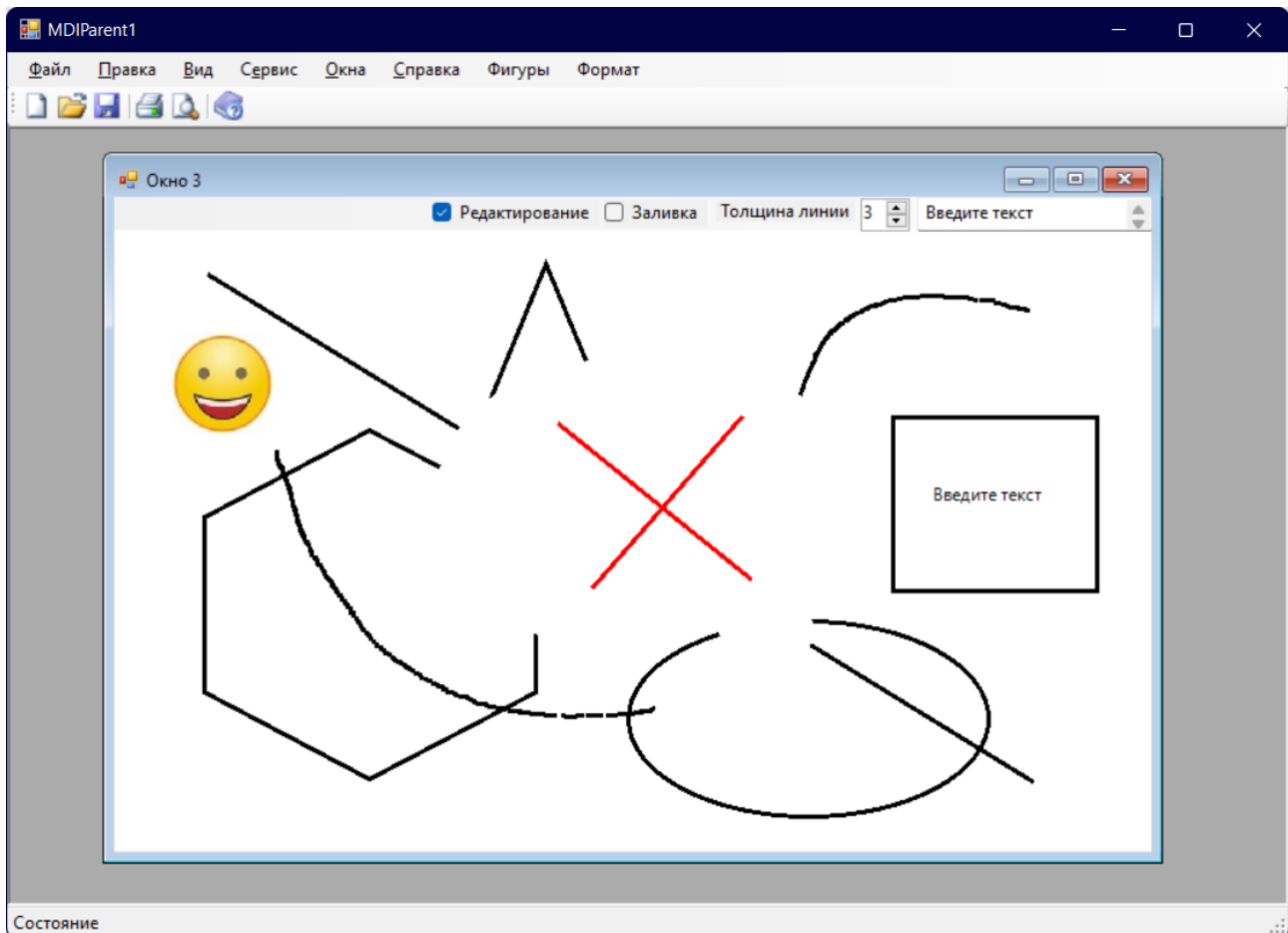


Рисунок 9 – Изменение цвета линии

- 2) Цвет заливки – устанавливает выбранный в диалоге выбора цвета цвет заливки
- 3) Цвет текста – устанавливает выбранный в диалоге выбора цвета цвет текста
- 4) Формат текста – устанавливает выбранный в диалоге выбора формата текста текст согласно рисунка 10.

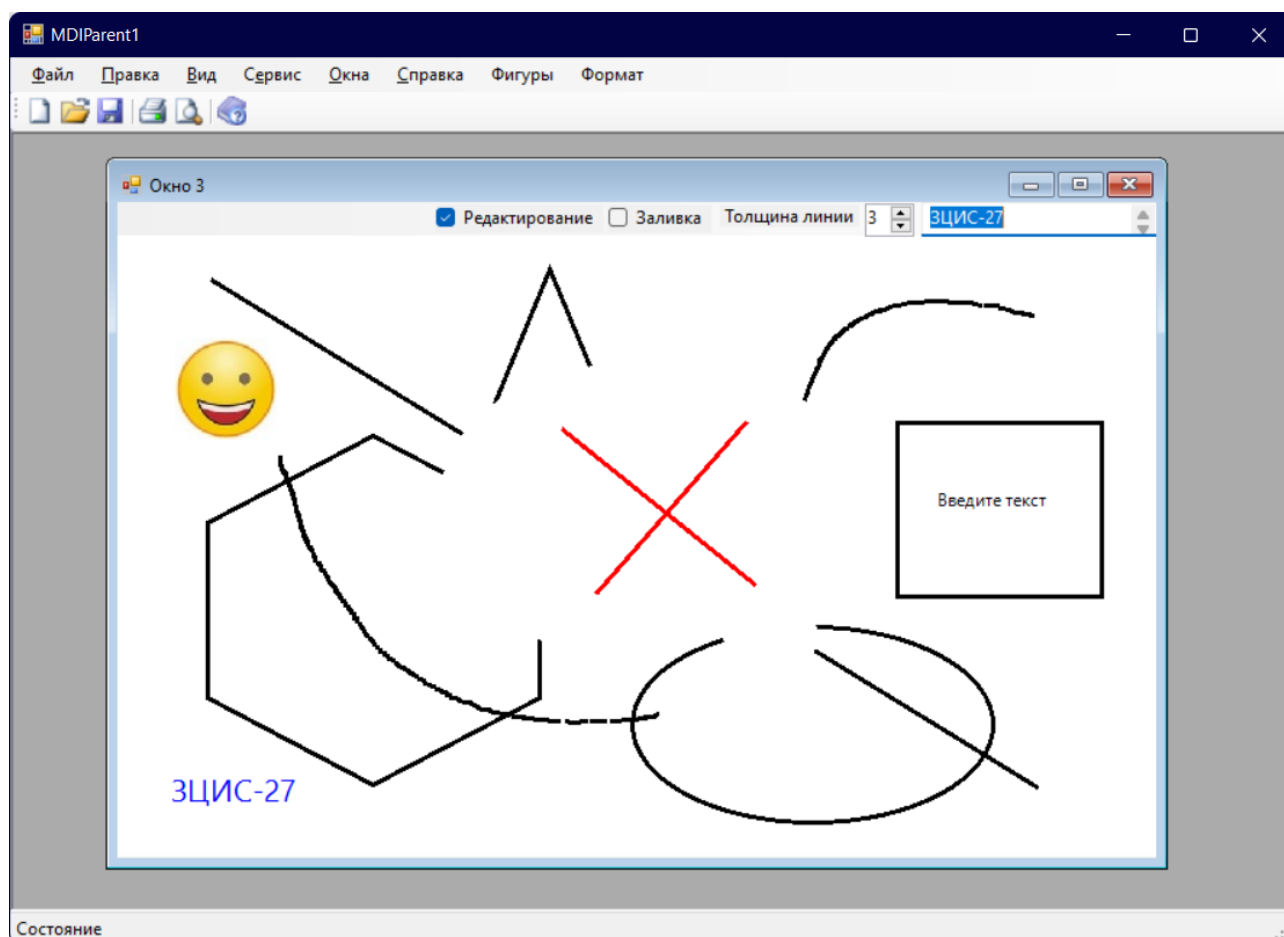


Рисунок 10 – Изменение формата текста

2.2.2 Панель инструментов

Панель инструментов предназначена для удобства доступа к часто используемым функциям, дублируя соответствующие элементы главного меню.

2.2.3 Оконное меню

Управляет функциями данного активного окна:

а) Редактирование – переключает окно из режима просмотра согласно рисунка 11 в режим редактирования согласно рисунка 12 и обратно

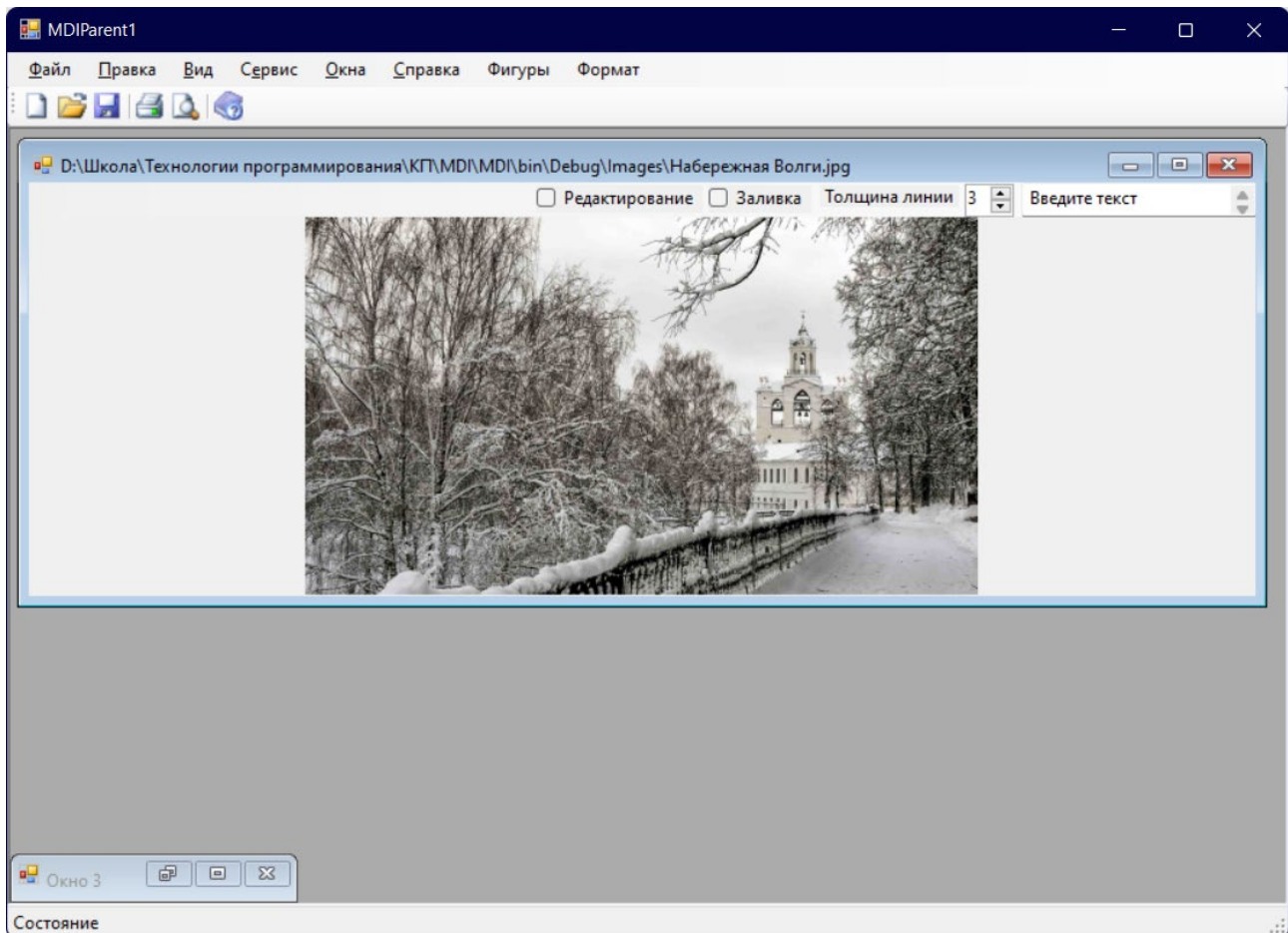


Рисунок 11 – Режим просмотра

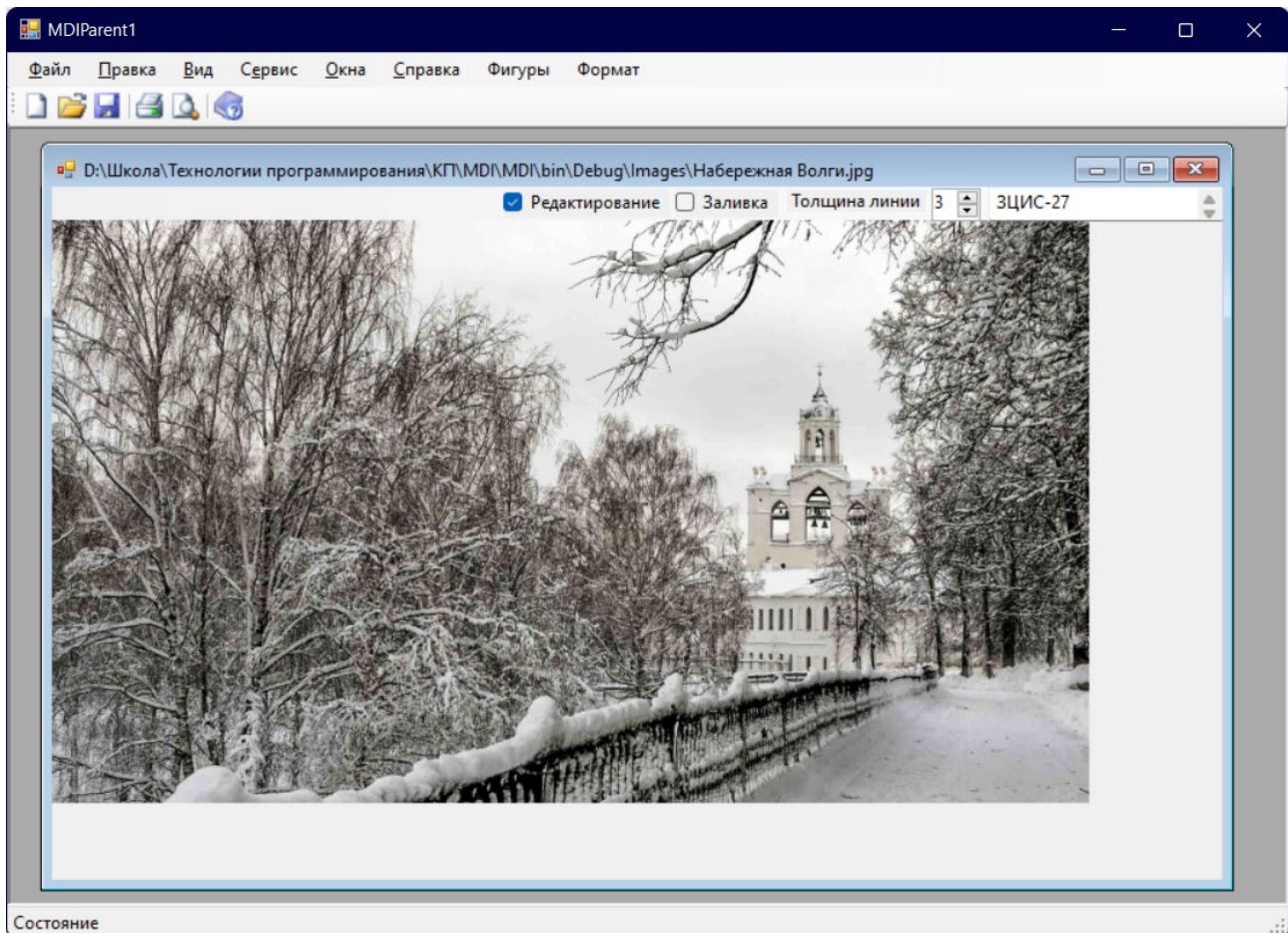


Рисунок 12 – Режим редактирования

- б) Заливка – включает и отключает режим заливки замкнутой фигуры
- в) Толщина линии – служит для указания требуемой толщины линии согласно рисунка 13

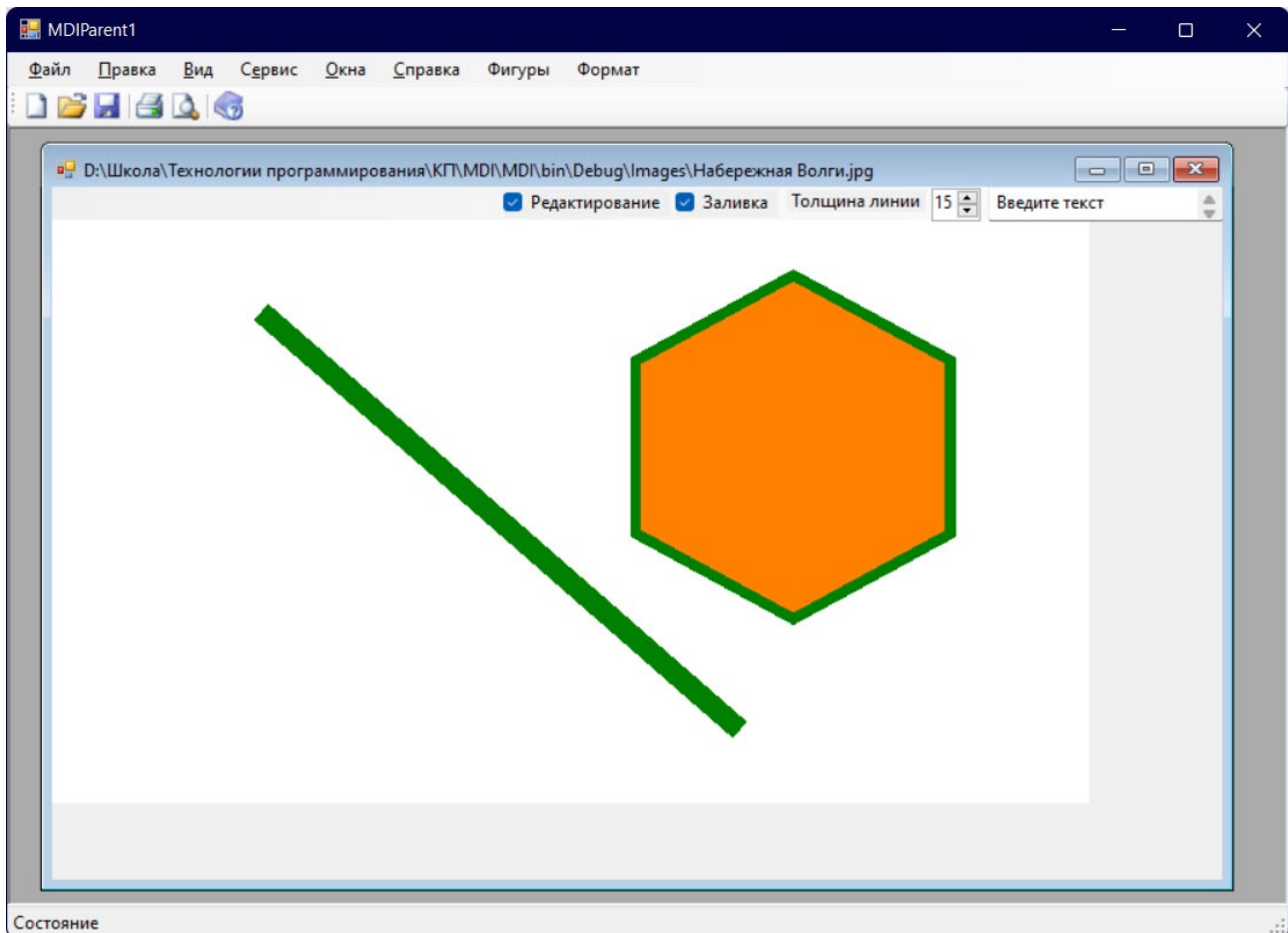


Рисунок 13 – Изменение толщины линии

г) Текстовая панель – предназначена для предварительного ввода текста.

Приложение А

Код программы

MDIParent1.cs

```

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;
using System.IO;

namespace MDI
{
    public partial class MDIParent1 : Form
    {
        private int childFormNumber = 0;
        public Bitmap bitCom;
        public string fileName;

        public MDIParent1()
        {
            InitializeComponent();
        }

        private void ShowNewForm(object sender, EventArgs e)
        {
            Form1 childForm = new Form1();
            childForm.MdiParent = this;
            childForm.Text = "Окно " + childFormNumber++;
            childForm.Show();
        }

        private void OpenFile(object sender, EventArgs e) // Открыть
        {
            Form1 curForm = (Form1)this.ActiveMdiChild;
            OpenFileDialog openFileDialog = new OpenFileDialog();
            openFileDialog.Filter = "Файлы изображений (*.jpg)|*.jpg|Файлы изображений (*.jpeg)|*.jpeg|Файлы изображений (*.bmp)|*.bmp|Файлы изображений (*.gif)|*.gif|Файлы изображений (*.png)|*.png|Все файлы (*.*)|*.*";
            if (openFileDialog.ShowDialog(this) == DialogResult.OK)
            {
                if (this.ActiveMdiChild == null)
                {
                    fileName = openFileDialog.FileName;
                    Form1 childForm = new Form1();
                    childForm.MdiParent = this;
                    childForm.pictureBox1.Image = Image.FromFile(fileName);
                    childForm.pictureBox1.SizeMode = PictureBoxSizeMode.Zoom;
                    childForm.bitmap = new Bitmap(childForm.pictureBox1.Image);
                    childForm.pictureBox1.Image = (Image)childForm.bitmap;
                    childForm.graphics1 = childForm.pictureBox1.CreateGraphics();
                    childForm.graphics2 = Graphics.FromImage(childForm.pictureBox1.Image);
                    childForm.Text = openFileDialog.FileName;
                    childForm.Show();
                    childForm.checkBox2.CheckState = CheckState.Unchecked;
                }
            }
            else
            {
            }
        }
    }
}

```

```

        fileName = openFileDialog.FileName;
        curForm.pictureBox1.Image = Image.FromFile(fileName);
        curForm.pictureBox1.SizeMode = PictureBoxSizeMode.Zoom;
        curForm.bitmap = new Bitmap(curForm.pictureBox1.Image);
        curForm.pictureBox1.Image = (Image)curForm.bitmap;
        curForm.graphics1 = curForm.pictureBox1.CreateGraphics();
        curForm.graphics2 = Graphics.FromImage(curForm.pictureBox1.Image);
        curForm.Text = fileName;
        curForm.checkBox2.CheckState = CheckState.Unchecked;
    }
}

Как private void SaveAsToolStripMenuItem_Click(object sender, EventArgs e) // Сохранить
{
    SaveFileDialog saveFileDialog = new SaveFileDialog();
    saveFileDialog.Filter = "Файлы изображений (*.jpg)|*.jpg|Файлы изображений (*.jpeg)|*.jpeg|Файлы изображений (*.bmp)|*.bmp|Файлы изображений (*.gif)|*.gif|Файлы изображений (*.png)|*.png|Все файлы (*.*)|*.*";
    if (saveFileDialog.ShowDialog(this) == DialogResult.OK)
    {
        string fileName = saveFileDialog.FileName;
        Form1 curForm = (Form1)this.ActiveMdiChild;
        curForm.bitmap = new Bitmap(curForm.pictureBox1.Image);
        curForm.bitmap.Save(fileName, System.Drawing.Imaging.ImageFormat.Bmp);
        curForm.Text = fileName;
    }
}

private void saveToolStripMenuItem_Click(object sender, EventArgs e) // Сохранить
{
    try
    {
        Form1 curForm = (Form1)this.ActiveMdiChild;
        fileName = curForm.Text;
        if (fileName.Contains("Окно"))
        {
            SaveFileDialog saveFileDialog = new SaveFileDialog();
            saveFileDialog.Filter = "Файлы изображений (*.jpg)|*.jpg|Файлы изображений (*.jpeg)|*.jpeg|Файлы изображений (*.bmp)|*.bmp|Файлы изображений (*.gif)|*.gif|Файлы изображений (*.png)|*.png|Все файлы (*.*)|*.*";
            if (saveFileDialog.ShowDialog(this) == DialogResult.OK)
            {
                string fileName = saveFileDialog.FileName;
                curForm.bitmap = new Bitmap(curForm.pictureBox1.Image);
                curForm.bitmap.Save(fileName, System.Drawing.Imaging.ImageFormat.Bmp);
                curForm.Text = fileName;
            }
        }
        curForm.bitmap = new Bitmap(curForm.pictureBox1.Image);
        curForm.bitmap.Save(fileName, System.Drawing.Imaging.ImageFormat.Bmp);
    }
    catch
    {
        MessageBox.Show("Внесите изменения.");
    }
}

private void saveToolStripButton_Click(object sender, EventArgs e) // Сохранить
{
    try
    {
        Form1 curForm = (Form1)this.ActiveMdiChild;

```

```

        fileName = curForm.Text;
        if(fileName.Contains("Окно"))
        {
            SaveFileDialog saveFileDialog = new SaveFileDialog();
            saveFileDialog.Filter = "Файлы изображений (*.jpg)|*.jpg|Файлы
изображений (*.jpeg)|*.jpeg|Файлы изображений (*.bmp)|*.bmp|Файлы изображений
(*.gif)|*.gif|Файлы изображений (*.png)|*.png|Все файлы (*.*)|*.*";
            if (saveFileDialog.ShowDialog(this) == DialogResult.OK)
            {
                string fileName = saveFileDialog.FileName;
                curForm.bitmap = new Bitmap(curForm.pictureBox1.Image);
                curForm.bitmap.Save(fileName,
System.Drawing.Imaging.ImageFormat.Bmp);
                curForm.Text = fileName;
            }
        }
        curForm.textBox1.Text = fileName;
        curForm.bitmap = new Bitmap(curForm.pictureBox1.Image);
        curForm.bitmap.Save(fileName, System.Drawing.Imaging.ImageFormat.Bmp);
    }
    catch
    {
        MessageBox.Show("Внесите изменения.");
    }
}

private void ExitToolsStripMenuItem_Click(object sender, EventArgs e)
{
    this.Close();
}

private void выделитьToolStripMenuItem_Click(object sender, EventArgs e) // Выделить
{
    Form1 curForm = (Form1)this.ActiveMdiChild;
    curForm.tl = 2;
    curForm.pen.Width = 1;
    curForm.penCut = true;
}

private void selectAllToolStripMenuItem_Click(object sender, EventArgs e) //
Выделить Все
{
    Form1 curForm = (Form1)this.ActiveMdiChild;
    curForm.x = 1;
    curForm.y = 1;
    curForm.a = curForm.pictureBox1.Width - 2;
    curForm.b = curForm.pictureBox1.Height - 2;
    curForm.penCut = true;
}

private void CutToolStripMenuItem_Click(object sender, EventArgs e) //Вырезать
{
    Form1 curForm = (Form1)this.ActiveMdiChild;
    try
    {
        if ((curForm.a != 0) | (curForm.b != 0) | (curForm.x != 0) | (curForm.y !=
0))
        {
            int i, j;
            bitCom = new Bitmap(curForm.a, curForm.b);
            for (i = curForm.x + 1; i < curForm.x + curForm.a - 1; i++)
            {
                for (j = curForm.y + 1; j < curForm.y + curForm.b - 1; j++)
                {

```



```

        Color color1 = curForm.bitmap.GetPixel(i, j);
        bitCom.SetPixel(i - curForm.x, j - curForm.y, color1);
    }
}

    int k, l;
    for (k = curForm.x - 1; k < curForm.x + curForm.a + 1; k++)
        for (l = curForm.y - 1; l < curForm.y + curForm.b + 1; l++)
        {
            curForm.bitmap.SetPixel(k, l, Color.FromArgb(255, 255, 255,
255));
        }
    }
}
catch
{
    MessageBox.Show("Выделите область.");
}
Refresh();
curForm.penCut = false;
curForm.a = 0; curForm.b = 0; curForm.x = 0; curForm.y = 0;
}

private void CopyToolStripMenuItem_Click(object sender, EventArgs e) //Копировать
{
    Form1 curForm = (Form1)this.ActiveMdiChild;
    try
    {
        if ((curForm.a != 0) | (curForm.b != 0) | (curForm.x != 0) | (curForm.y !=
0))
        {
            int i, j;
            bitCom = new Bitmap(curForm.a, curForm.b);
            for (i = curForm.x + 1; i < curForm.x + curForm.a - 1; i++)
            {
                for (j = curForm.y + 1; j < curForm.y + curForm.b - 1; j++)
                {
                    Color color1 = curForm.bitmap.GetPixel(i, j);
                    bitCom.SetPixel(i - curForm.x, j - curForm.y, color1);
                }
            }
        }
    }
    catch
    {
        MessageBox.Show("Выделите область.");
    }
    curForm.penCut = false;
    curForm.a = 0; curForm.b = 0; curForm.x = 0; curForm.y = 0;
}

private void PasteToolStripMenuItem_Click(object sender, EventArgs e) // Вставить
{
    Form1 curForm = (Form1)this.ActiveMdiChild;
    if ((bitCom != null) && curForm.checkBox2.Checked)
        curForm.graphics2.DrawImage(bitCom, curForm.x, curForm.y);
    curForm.pictureBox1.Refresh();
}

private void ToolBarToolStripMenuItem_Click(object sender, EventArgs e)
{
    toolStrip.Visible = toolBarToolStripMenuItem.Checked;
}

```

```

private void StatusBarToolStripMenuItem_Click(object sender, EventArgs e)
{
    statusStrip.Visible = statusBarToolStripMenuItem.Checked;
}

private void CascadeToolStripMenuItem_Click(object sender, EventArgs e)
{
    LayoutMdi(MdiLayout.Cascade);
}

private void TileVerticalToolStripMenuItem_Click(object sender, EventArgs e)
{
    LayoutMdi(MdiLayout.TileVertical);
}

private void TileHorizontalToolStripMenuItem_Click(object sender, EventArgs e)
{
    LayoutMdi(MdiLayout.TileHorizontal);
}

private void ArrangeIconsToolStripMenuItem_Click(object sender, EventArgs e)
{
    LayoutMdi(MdiLayout.ArrangeIcons);
}

private void CloseAllToolStripMenuItem_Click(object sender, EventArgs e)
{
    foreach (Form childForm in MdiChildren)
    {
        childForm.Close();
    }
}

private void aboutToolStripMenuItem_Click(object sender, EventArgs e) // 0 программе
{
    MessageBox.Show("Технологии программирования\nКурсовая работа\nВыполнил студент
группы ДЦИС-27\nКонторин П.Л.");
}

private void toolStripMenuItem1_Click(object sender, EventArgs e) // Очистить
{
    Form1 curForm = (Form1)this.ActiveMdiChild;
    curForm.Plotno();
    curForm.checkBox2.CheckState = CheckState.Unchecked;
}
}
}

```

Form1.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;

```

```

namespace MDI
{
    public partial class Form1 : Form
    {
        public Pen pen;
        public Bitmap bitmap, bitmap1;
        public Font textFont;
        public int tl = 0;
        public int x, y;
        public int a, b;
        public bool penCut = false;
        double k, l;
        public Graphics graphics1, graphics2;
        SolidBrush brush, brushtext, brushcur, brrez;
        public Form1()
        {
            InitializeComponent();
            Polotno();
        }

        public void Polotno() // Полотно
        {
            bitmap = new Bitmap(pictureBox1.Width, pictureBox1.Height);
            int i, j;
            for (i = 0; i < pictureBox1.Width; i++)
                for (j = 0; j < pictureBox1.Height; j++)
                {
                    bitmap.SetPixel(i, j, Color.FromArgb(255, 255, 255, 255));
                }
            pictureBox1.Image = (Image)bitmap;
            graphics1 = pictureBox1.CreateGraphics();
            graphics2 = Graphics.FromImage(pictureBox1.Image);
            pen = new Pen(Color.Black);
            pen.Width = Convert.ToInt16(numericUpDown1.Text);
            brush = new SolidBrush(Color.Black);
            brushtext = new SolidBrush(Color.Black);
            brushcur = new SolidBrush(Color.Black);
            brrez = new SolidBrush(Color.White);
            textFont = new Font("Segoe UI", 9);
            tl = 0;
            x = 0;
            y = 0;
            pictureBox1.ContextMenuStrip = contextMenuStrip1;
            textBox1.Text = "Введите текст";
            textBox1.ForeColor = Color.Black;
            textBox1.Font = new Font("Segoe UI", 9);
            цветЛинииToolStripMenuItem.BackColor = Color.White;
            цветЗаливкиToolStripMenuItem.BackColor = Color.White;
            цветТекстаToolStripMenuItem.BackColor = Color.White;
        }

        private void checkBox2_Click(object sender, EventArgs e) // Редактировать
        {
            if (checkBox2.Checked)
            {
                k = Convert.ToDouble(bitmap.Width) / Convert.ToDouble(pictureBox1.Width);
                l = Convert.ToDouble(bitmap.Height) / Convert.ToDouble(pictureBox1.Height);
                Size size = new Size();
                if (k > 1)
                {
                    size.Width = Convert.ToInt16(bitmap.Width / k);
                    size.Height = Convert.ToInt16(bitmap.Height / k);
                }
            }
        }
    }
}

```

```

    }
    else
    {
        size.Width = Convert.ToInt16(bitmap.Width / 1);
        size.Height = Convert.ToInt16(bitmap.Height / 1);
    }
    bitmap = new Bitmap((Image)bitmap, size);
    pictureBox1.Dock = DockStyle.None;
    pictureBox1.Size = size;
    pictureBox1.Image = (Image)bitmap;
    graphics1 = pictureBox1.CreateGraphics();
    graphics2 = Graphics.FromImage(pictureBox1.Image);
}
else
{
    pictureBox1.Dock = DockStyle.Fill;
}
}

private void вырезатьToolStripMenuItem_Click(object sender, EventArgs e) // Вырезать
{
    MDIParent1 parForm = (MDIParent1)this.MdiParent;
    try
    {
        if ((a != 0) | (b != 0) | (x != 0) | (y != 0))
        {
            int i, j;
            parForm.bitCom = new Bitmap(a, b);
            for (i = x + 1; i < x + a - 1; i++)
            {
                for (j = y + 1; j < y + b - 1; j++)
                {
                    Color color1 = bitmap.GetPixel(i, j);
                    parForm.bitCom.SetPixel(i - x, j - y, color1);
                }
            }
            int k, l;
            for (k = x - 1; k < x + a + 1; k++)
            for (l = y - 1; l < y + b + 1; l++)
            {
                bitmap.SetPixel(k, l, Color.FromArgb(255, 255, 255, 255));
            }
        }
        catch
        {
            MessageBox.Show("Выделите область.");
        }
        Refresh();
        penCut = false;
        a = 0; b = 0; x = 0; y = 0;
    }
}

private void копироватьToolStripMenuItem_Click(object sender, EventArgs e) //
Копировать
{
    MDIParent1 parForm = (MDIParent1)this.MdiParent;
    try
    {
        if ((a != 0) | (b != 0) | (x != 0) | (y != 0))
        {
            int i, j;
            parForm.bitCom = new Bitmap(a, b);
            for (i = x + 1; i < x + a - 1; i++)

```

```

        {
            for (j = y + 1; j < y + b - 1; j++)
            {
                Color color1 = bitmap.GetPixel(i, j);
                parForm.bitCom.SetPixel(i - x, j - y, color1);
            }
        }
    }
}
catch
{
    MessageBox.Show("Выделите область.");
}
penCut = false;
a = 0; b = 0; x = 0; y = 0;
}

private void вставитьToolStripMenuItem_Click_1(object sender, EventArgs e) //
Вставить
{
    MDIParent1 parForm = (MDIParent1)this.MdiParent;
    if ((parForm.bitCom != null) && checkBox2.Checked)
        graphics2.DrawImage(parForm.bitCom, x, y);
    pictureBox1.Refresh();
}

private void цветЛинииToolStripMenuItem_Click(object sender, EventArgs e) // Цвет
Линии
{
    if (colorDialog1.ShowDialog() == DialogResult.OK)
    {
        pen.Color = colorDialog1.Color;
        brushcur.Color = colorDialog1.Color;
        цветЛинииToolStripMenuItem.BackColor = colorDialog1.Color;
    }
}

private void цветЗаливкиToolStripMenuItem_Click(object sender, EventArgs e) //Цвет
заливки
{
    if (colorDialog1.ShowDialog() == DialogResult.OK)
    {
        brush.Color = colorDialog1.Color;
        this.цветЗаливкиToolStripMenuItem.BackColor = colorDialog1.Color;
    }
}

private void цветТекстаToolStripMenuItem_Click(object sender, EventArgs e) // Цвет
Текста
{
    if (colorDialog1.ShowDialog() == DialogResult.OK)
    {
        brushtext.Color = colorDialog1.Color;
        this.цветТекстаToolStripMenuItem.BackColor = colorDialog1.Color;
    }
}

private void форматТекстаToolStripMenuItem_Click(object sender, EventArgs e) //
Формат Текста
{
    if (fontDialog1.ShowDialog() == DialogResult.OK)
    {
        textFont = fontDialog1.Font;
    }
}

```

```

    }
}

private void textBox1_MouseDown(object sender, MouseEventArgs e)
{
    textBox1.Text = null;
}

private void линияToolStripMenuItem_Click(object sender, EventArgs e) // Линия
{
    t1 = 0;
}

private void криваяToolStripMenuItem_Click(object sender, EventArgs e) // Кривая
{
    t1 = 1;
}

private void прямоугольникToolStripMenuItem_Click(object sender, EventArgs e) //
Прямоугольник
{
    t1 = 2;
}

private void эллипсToolStripMenuItem_Click(object sender, EventArgs e) // Эллипс
{
    t1 = 3;
}

private void ластикToolStripMenuItem_Click(object sender, EventArgs e) // Ластик
{
    t1 = 4;
}

private void текстToolStripMenuItem_Click(object sender, EventArgs e) // Текст
{
    t1 = 5;
}

private void треугольникToolStripMenuItem_Click(object sender, EventArgs e) //
Треугольник
{
    t1 = 6;
}

private void изображениеToolStripMenuItem_Click(object sender, EventArgs e) //
Изображение
{
    t1 = 7;
}

private void шестиугольникToolStripMenuItem_Click(object sender, EventArgs e) //
Шестиугольник
{
    t1 = 8;
}

private void pictureBox1_MouseDown_1(object sender, MouseEventArgs e) // Мышь Вниз
{
    if (e.Button == MouseButtons.Left)
    {
        x = e.X;
        y = e.Y;
    }
}

```

```

        if ((e.Button == MouseButton.Right) && (penCut == false))
        {
            x = e.X;
            y = e.Y;
        }
        if (penCut == true) pen.Width = 1;
        else pen.Width = Convert.ToInt16(numericUpDown1.Text);
    }

    private void pictureBox1_MouseMove(object sender, MouseEventArgs e) // Мышь Двигать
    {
        if ((e.Button == MouseButton.Left) && checkBox2.Checked)
            if (e.Button == MouseButton.Left)
            {
                int x1 = e.X, y1 = e.Y; switch (t1)
                {
                    case 0: graphics1.DrawLine(pen, x, y, x1, y1); break;
                    case 1: graphics2.FillRectangle(brushcur, x1, y1, 4, 4); break;
                    case 2: graphics1.DrawRectangle(pen, x, y, x1 - x, y1 - y); break;
                    case 3: graphics1.DrawEllipse(pen, x, y, x1 - x, y1 - y); break;
                    case 4: graphics2.FillEllipse(brrez, x1, y1, 30, 30); break;
                    case 5: break;
                    case 6:
                        {
                            PointF point1 = new PointF(x, y1);
                            PointF point2 = new PointF((x + x1) / 2, y);
                            PointF point3 = new PointF(x1, y1);
                            PointF[] curvePoints = { point1, point2, point3 };
                            graphics1.DrawPolygon(pen, curvePoints);
                        }
                        break;
                    case 7: break;
                    case 8:
                        {
                            try
                            {
                                int a = Convert.ToUInt16((x1 - x) * 0.9375);
                                int a1 = Convert.ToUInt16((x1 - x) * 0.0625);
                                int b = Convert.ToUInt16((y1 - y) * 0.25);
                                int b1 = Convert.ToUInt16((y1 - y) * 0.75);
                                PointF point1 = new PointF((x + x1) / 2, y);
                                PointF point2 = new PointF((x + a), (y + b));
                                PointF point3 = new PointF((x + a), (y + b1));
                                PointF point4 = new PointF((x + x1) / 2, y1);
                                PointF point5 = new PointF((x + a1), (y + b1));
                                PointF point6 = new PointF((x + a1), (y + b));
                                PointF[] curvePoints = { point1, point2, point3, point4,
point5, point6 };

                                graphics1.DrawPolygon(pen, curvePoints);
                            }
                            catch
                            {
                                MessageBox.Show("Измените значение.");
                            }
                        }
                        break;
                    default: graphics1.DrawLine(pen, x, y, x1, y1); break;
                }
            }
        }
        pictureBox1.Refresh();
    }

    private void pictureBox1_MouseUp_1(object sender, MouseEventArgs e) // Мышь вверх

```

```

{
    if ((e.Button == MouseButtons.Left)&&(checkBox2.Checked))
    {
        int x1 = e.X, y1 = e.Y;
        switch (t1)
        {
            case 0: graphics2.DrawLine(pen, x, y, x1, y1); break;
            case 1: break;
            case 2:
            {
                if (checkBox1.Checked)
                {
                    graphics2.DrawRectangle(pen, x, y, x1 - x, y1 - y);
                    graphics2.FillRectangle(brush, x, y, x1 - x, y1 - y);
                }
                else
                {
                    graphics2.DrawRectangle(pen, x, y, x1 - x, y1 - y);
                    a = x1 - x;
                    b = y1 - y;
                }
                break;
            }
            case 3:
            {
                if (checkBox1.Checked)
                {
                    graphics2.DrawEllipse(pen, x, y, x1 - x, y1 - y);
                    graphics2.FillEllipse(brush, x, y, x1 - x, y1 - y);
                }
                else
                {
                    graphics2.DrawEllipse(pen, x, y, x1 - x, y1 - y);
                }
                break;
            }
            case 4: break;
            case 5: graphics2.DrawString(textBox1.Text, textFont, brushtext, x, y);
            break;
            case 6:
            {
                if (checkBox1.Checked)
                {
                    PointF point1 = new PointF(x, y1);
                    PointF point2 = new PointF((x + x1) / 2, y);
                    PointF point3 = new PointF(x1, y1);
                    PointF[] curvePoints = { point1, point2, point3 };
                    graphics2.DrawPolygon(pen, curvePoints);
                    graphics2.FillPolygon(brush, curvePoints);
                }
                else
                {
                    PointF point1 = new PointF(x, y1);
                    PointF point2 = new PointF((x + x1) / 2, y);
                    PointF point3 = new PointF(x1, y1);
                    PointF[] curvePoints = { point1, point2, point3 };
                    graphics2.DrawPolygon(pen, curvePoints);
                }
            }
            break;
            case 7:
            {
                Image image = Image.FromFile(Directory.GetCurrentDirectory() +
"/images/image.jpg");
                graphics2.DrawImage(image, x, y);
            }
        }
    }
}

```