# Разработка MDI-приложения на платформе .NET

**Задание**

Разработать в среде Microsoft Visual Studio на языке С# MDI-приложение (Multiple Document Interface), позволяющее создавать и просматривать растровые изображения в форматах BMP, JPG и PNG.

В приложении необходимо реализовать следующие функции (нарастающим итогом).

|  |  |
| --- | --- |
| Оценка | Реализованные функции |
| 4-5 балла | * 1. команды приложения доступны через главное меню и панель инструментов, главное меню и панель инструментов контекстно зависима (например, если нет открытых окон, команды «Сохранить» и «Сохранить как…» должны быть неактивны.);   2. в меню и диалоговых окнах поддерживаются горячие клавиши;   3. в меню поддерживаются горячие сочетания клавиш;   4. рисование пером;   5. изменение цвета пера;   6. изменение толщины пера   7. команды автоматического расположения дочерних окон;   8. окно «О программе»   9. сохранение файла в формате BMP и JPG с использованием стандартных диалоговых окон, необходима поддержка фильтра по расширению. Необходимо реализовать команды **Сохранить как…** и **Сохранить**. При первом выполнении команды **Сохранить** для нового изображения запрашивается имя файла, при последующих имя файла не запрашивается.   10. загрузка файла в формате BMP и JPG с использованием стандартных диалоговых окон, необходима поддержка фильтра по расширению.   11. при выполнении всех операций реализована обработка исключительных ситуаций. |
| 6-7 баллов | 1. Инструмент «Линия», позволяющий рисовать линии; 2. Инструмент «Эллипс», позволяющий рисовать эллипсы; 3. Инструмент «Ластик», позволяющий стереть изображение в некотором радиусе точки; 4. При рисовании инструментами отображается промежуточный размер фигуры пока пользователь не отпусти кнопку мыши; 5. Поддерживается рисование закрашенных и не закрашенных фигур. 6. При смене инструмента курсор мыши в окне рисования должен меняться соответствующем образом. 7. При закрытии файла пользователю должно быть предложено сохранить измененный файл или отказаться от закрытия. Если файл не менялся, вопрос не задается. |
| 8 баллов | 1. Инструмент «Масштаб+», позволяющий увеличить масштаб изображения без искажения; 2. Инструмент «Масштаб-», позволяющий уменьшить масштаб изображения без искажения; 3. Инструмент «Текст»; 4. Инструмент закраски замкнутой области («Ведро с краской»). 5. Рисование фигуры по варианту (вариант указан в рабочей ведомости ЛМС) |
| 9 | Реализованы «причаливающие» окна и вкладки для документов ([DockPanel Suite](http://dockpanelsuite.com/) или подобная для WinForms), панель инструментов и меню реализовано с использованием ленты «Ribbon». |
| 10 баллов | Приложение реализовано на фрейморке отличном от WinForms (Windows Presentation Foundation, .NET MAUI и др.) |

**Вариант 1**

Инструмент «Звезда» позволяющий вставить в рисунок изображение n-конечной звезды (параметры фигуры (количество лучей, отношение внутреннего и внешнего радиуса) должны настраиваться).

**Вариант 2**

Инструмент «Правильный многоугольник» позволяющий вставить в рисунок изображение n-угольника (параметр n должен настраиваться).

**Вариант 3**

Инструмент «Выноска»

**Вариант 4**

Инструмент «Смайлик»

**Вариант 5**

Инструмент «Сердце»

**Вариант 6**

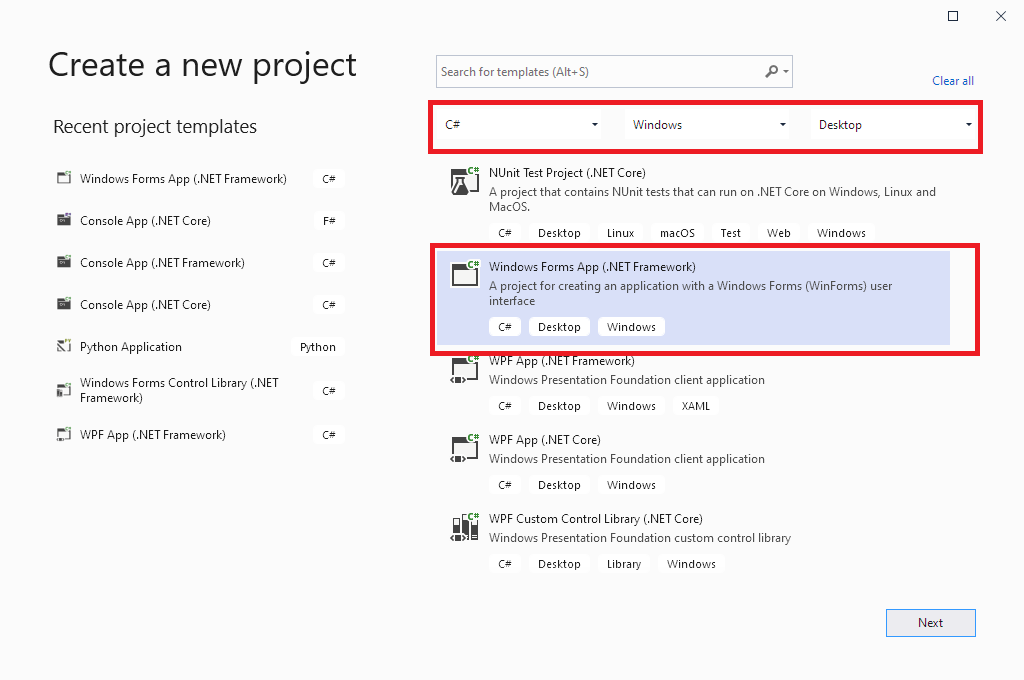
Инструмент «Стрелка»

**Вариант 7**

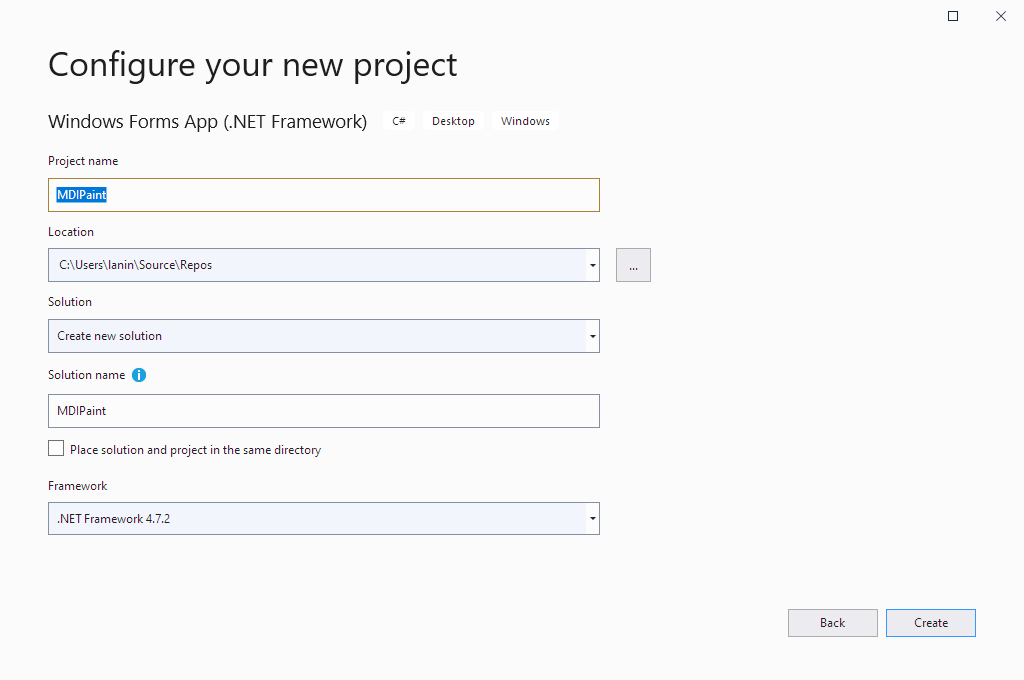
Инструмент «Сердце»

**Указания**

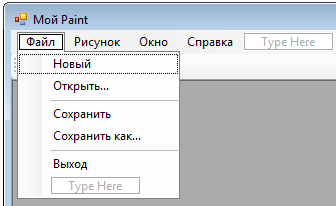
1. Создайте в Visual Studio новые проект Windows Forms App (.NET Framework). Для этого выберите пункт меню **File|New|Project…**. Для быстрого выбора укажите параметры фильтрации (C#, Windows, Desktop).

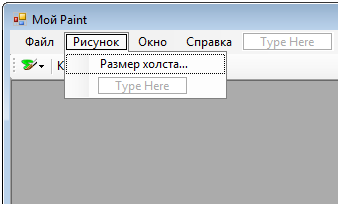


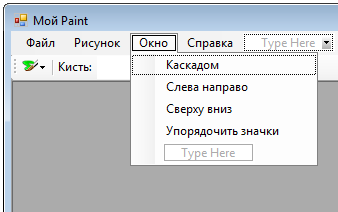
1. Назовите новый проект *MDIPaint*, укажите папку для размещения проекта при необходимости.

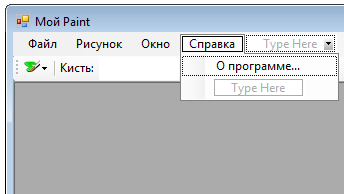


1. Для изменения свойств элементов экранных форм используется окно «Properties», которое отображается на экране при нажатии клавиши **F4** или выборе команды главного меню «View|Properties Window». Прежде чем изменить свойства какого-либо объекта его необходимо выделить щелчком мыши. Название выделенного объекта отображается в верхней части окна «Properties».
2. Переименуйте созданную по умолчанию форму Form1.cs. Выполните команду **Переименовать** в контекстном меню. Введите имя *MainForm*. Согласитесь переименовать зависимые объекты в появившемся сообщении.
3. Установите свойство **IsMdiContainer** равным *True*, для того чтобы сделать форму родительской. Используя свойство **Text** формы, измените заголовок окна.
4. Отобразите на экране окно «Toolbox», если оно не видно на экране. Это можно сделать, нажав комбинацию клавиш «Ctrl+Alt+X» или выбрав соответствующий пункт меню «View». Элементы управления распределены по логическим группам, просмотреть все стандартные элементы управления можно в разделе «All Windows Forms».
5. Добавьте на форму элемент MenuStrip. Создайте структуру меню, как показано на рисунке ниже.

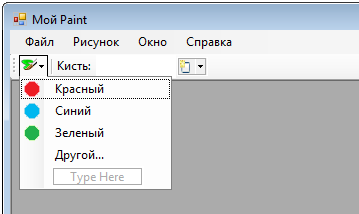




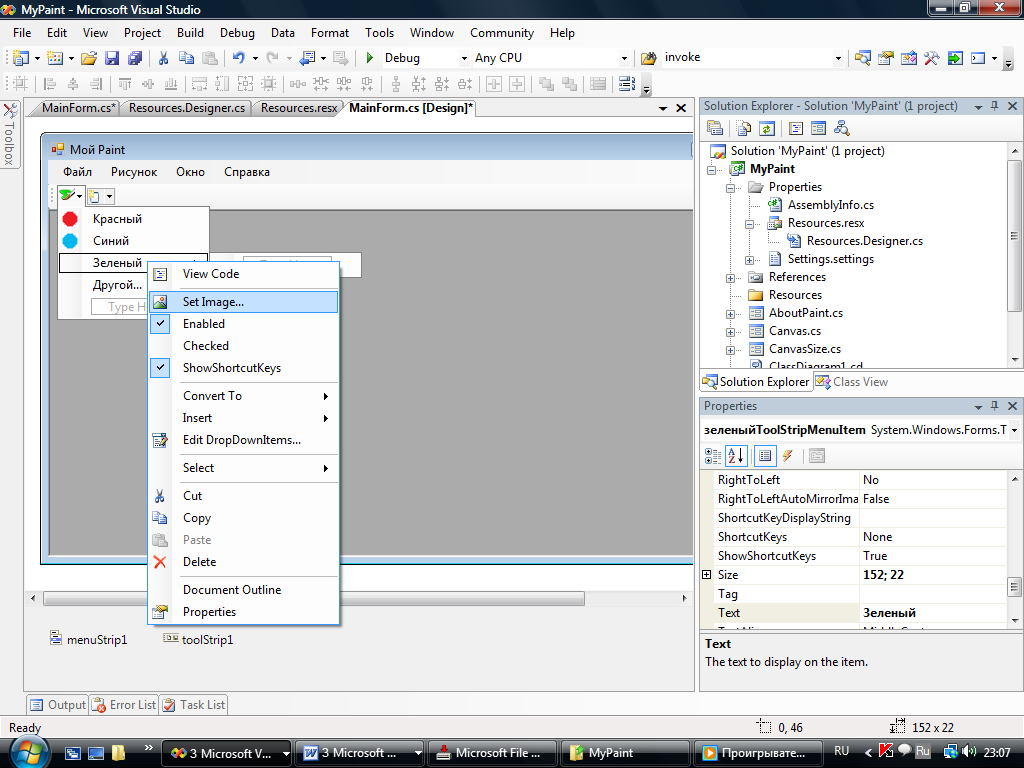


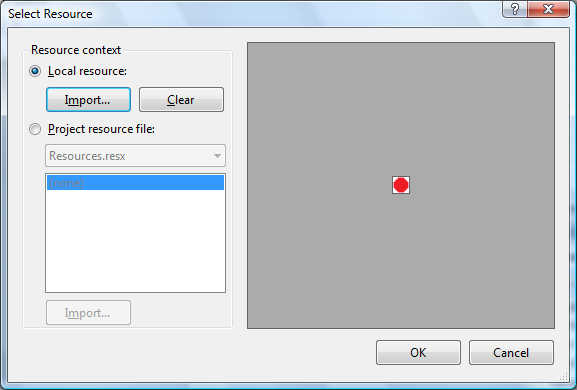


1. Настройте пунктам меню горячие клавиши и клавиатурные комбинации.
2. Выделите элемент управления MenuStrip и установите его свойство MdiWindowListItem равным *окноToolStripMenuItem* (имя пункта меню «Окно»). Использование данного свойства позволит автоматически отображать список дочерних окон в соответствующем пункте меню.
3. Добавьте элемент управления «ToolStrip», где поместите кнопки для выбора текущего инструмента рисования и цвета. Первым элементом установите DropDownButton, затем элемент Separator, затем элемент Label и TextBox.



1. Элементу DropDownButton добавьте четыре пункта для выбора цветов. Для установки пиктограммы используйте команду контекстного меню «Set Image…», используя команду «Import…» в разделе Local resource. Цвет маски рисунка задается с помощью свойства ImageTransparentColor каждого из элементов меню.





1. Добавьте обработчик команды меню «Выход».

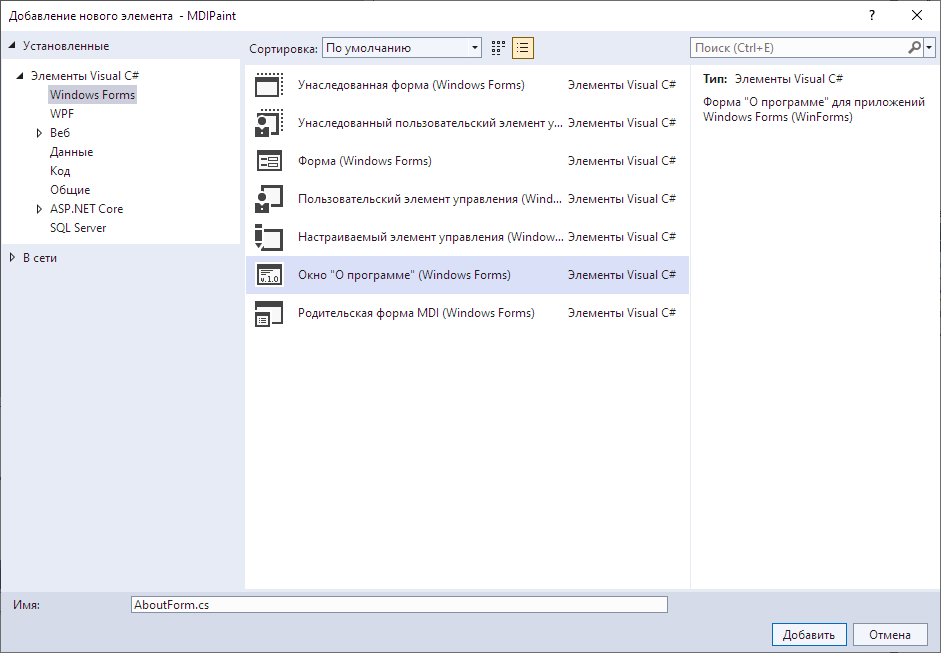
private void выходToolStripMenuItem\_Click(object sender, EventArgs e)

{

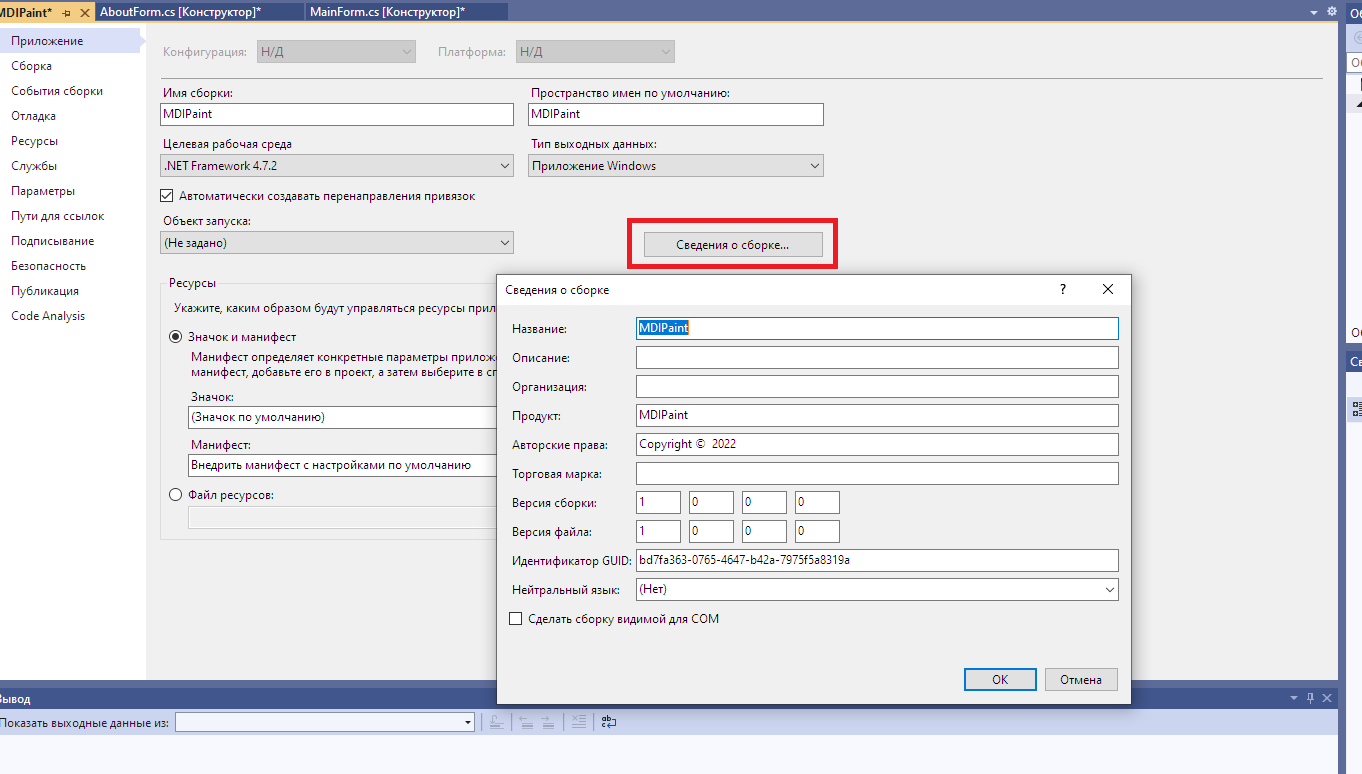
Application.Exit();

}

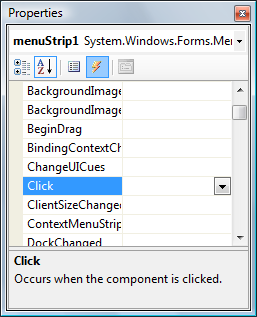
1. Выполните команду «Project|Add Windows Form…» еще раз. Выберите шаблон About Box и назовите новую форму *AboutForm*.



1. Выполните команду «Project|MyPaint Properties…». На вкладке «Application» нажмите кнопку «Assembly Information…» и заполните информацию о приложении. Введенная информация автоматически отобразится в окне «О программе…».



1. В главной форме выделите пункт меню «О программе». Перейдите в окно «Properties» и нажмите там кнопку «Events». В списке отобразятся все доступные события выделенного элемента управления, в данном случае элемента меню. Дважды щелкните мышкой по событию *Click* для создания обработчика события выбора элемента меню.



1. Введите представленный ниже программный код. Запустите приложение на выполнение.

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

{

var frmAbout = new AboutForm();

frmAbout.ShowDialog();

}

1. Добавьте код завершения программы.

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

{

Application.Exit();

}

1. Для добавления дочерней формы воспользуйтесь командой «Project|Add Windows Form…» в главном меню. Выберите шаблон Windows Form. Назовите дочернюю форму *DocumentForm*.
2. Установите свойство BackColor формы *DocumentForm* равным AppWorkspace, свойство AutoScroll равным True.
3. В программный код главной формы добавьте объявление параметров рисования и обработчик команды открытия нового документа.

public partial class MainForm : Form

{

public static Color Color { get; set; }

public static int Width { get; set; }

public MainForm()

{

InitializeComponent();

Color = Color.Black;

Width = 3;

}

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

{

var frmAbout = new AboutForm();

frmAbout.ShowDialog();

}

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

{

var frm = new DocumentForm();

frm.MdiParent = this;

frm.Show();

}

}

1. Добавьте следующий программный код в дочернюю форму.

public partial class DocumentForm : Form

{

private int x, y;

private Bitmap bitmap;

public DocumentForm()

{

InitializeComponent();

bitmap = new Bitmap(300, 200);

}

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

{

x = e.X;

y = e.Y;

}

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

{

if (e.Button == MouseButtons.Left)

{

Graphics g = Graphics.FromImage(bitmap);

g.DrawLine(new Pen(MainForm.Color, MainForm.Width), x, y, e.X, e.Y);

Invalidate();

x = e.X;

y = e.Y;

}

}

protected override void OnPaint(PaintEventArgs e)

{

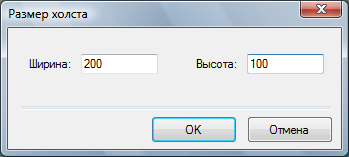
base.OnPaint(e);

e.Graphics.DrawImage(bitmap, 0, 0);

}

}

1. Установите форме DocumentForm свойство DoubleBuffered равное True для устранения мерцания при перерисовке.
2. Добавьте в проект новую форму с именем CanvasSizeForm. Для того чтобы данное окно отображалось как диалоговое окно, установите значения свойств MinimizeBox и MaximizeBox равными false, свойству FormBorderStyle присвойте значение FixedDialog, а свойству StartPosition – CenterParent. Разместите элементы управления как показано на рисунке. Для задания действий по умолчанию на нажатие клавиш Enter и Esc установите свойства формы AcceptButton равное btnOK, а свойство CancelButton равное btnCancel. Для кнопки btnOK установите свойство DialogResalt равное OK.



1. В главной форме добавьте обработчик события на пункт меню «Рисунок».

private void рисунокToolStripMenuItem\_DropDownOpening(object sender, EventArgs e)

{

размерХолстаToolStripMenuItem.Enabled=!(ActiveMdiChild==null);

}

1. В главной форме добавьте обработчик события на пункт меню «Размер холста…», данная команда должна вызывать изменение размера рисунка активного документа.
2. В главной форме добавьте возможность выбора цвета и размера кисти.

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

{

Color = Color.Red;

}

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

{

Color = Color.Blue;

}

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

{

Color = Color.Green;

}

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

{

ColorDialog cd=new ColorDialog();

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

Color = cd.Color;

}

­

1. На основе примера кода реализуйте команду **Открыть…**.

OpenFileDialog dlg = new OpenFileDialog();

dlg.Filter = "Windows Bitmap (\*.bmp)|\*.bmp| Файлы JPEG (\*.jpeg, \*.jpg)|\*.jpeg;\*.jpg|Все файлы ()\*.\*|\*.\*";

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

{

…

}

}

1. Добавьте команды автоматического расположения дочерних окон.

LayoutMdi(MdiLayout.Cascade);

LayoutMdi(MdiLayout.ArrangeIcons);

LayoutMdi(MdiLayout.TileHorizontal);

LayoutMdi(MdiLayout.TileVertical);

1. Примерный код рисования инструментом Линия представлен ниже.

using System.Drawing;

using System.Windows.Forms;

namespace MDIPaint

{

public partial class DocumentForm : Form

{

private int x, y;

private Bitmap bitmap;

public DocumentForm()

{

InitializeComponent();

bitmap = new Bitmap(300, 200);

}

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

{

x = e.X;

y = e.Y;

}

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

{

if (e.Button == MouseButtons.Left)

{

Refresh();

Graphics g = CreateGraphics();

g.DrawLine(new Pen(MainForm.Color, MainForm.Width), x, y, e.X, e.Y);

}

}

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

{

Graphics g = Graphics.FromImage(bitmap);

g.DrawLine(new Pen(MainForm.Color, MainForm.Width), x, y, e.X, e.Y);

x = e.X;

y = e.Y;

Invalidate();

}

protected override void OnPaint(PaintEventArgs e)

{

base.OnPaint(e);

e.Graphics.DrawImage(bitmap, 0, 0);

}

}

}