# 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.

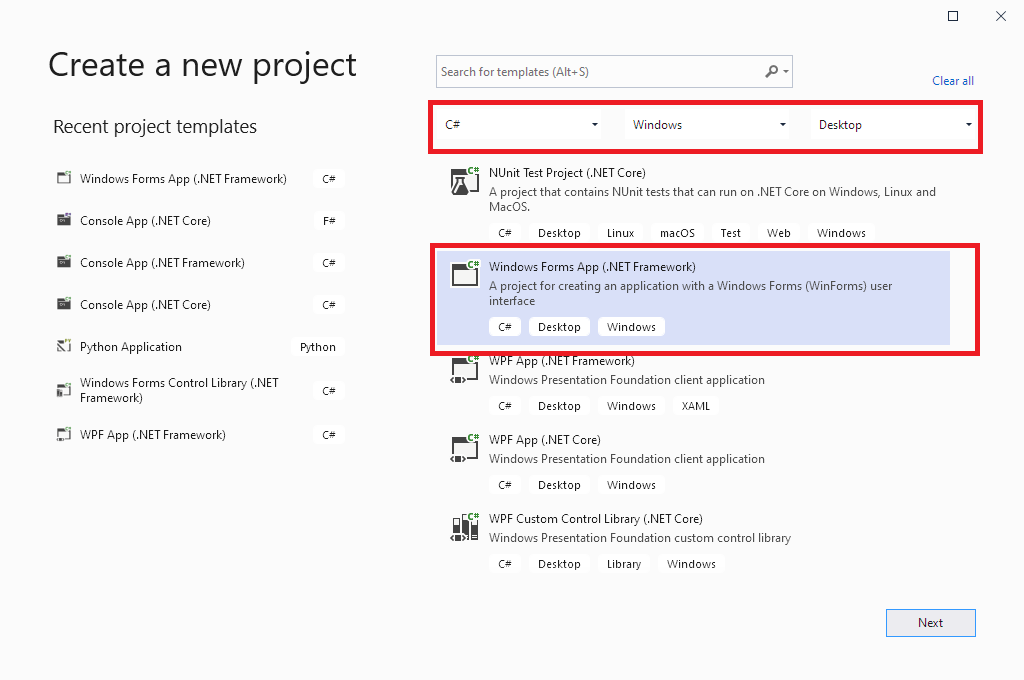
В приложении необходимо реализовать следующие функции:

1. **Редактирования изображения**
   1. рисование пером;
   2. изменение цвета пера;
   3. изменение толщины пера.
   4. инструмент «Звезда», позволяющий вставить в рисунок изображение n-конечной звезды в заданной позиции.
   5. Инструмент «Линия», позволяющий рисовать линии;
   6. Инструмент «Эллипс», позволяющий рисовать эллипсы;
   7. Инструмент «Ластик», позволяющий стереть изображение в некотором радиусе точки;
   8. Инструмент «Масштаб+», позволяющий увеличить масштаб изображения;
   9. Инструмент «Масштаб-», позволяющий уменьшить масштаб изображения;
2. **Работа с файлом**
   1. создание нового файла;
   2. сохранение файла в формате BMP и JPG;
   3. загрузка файла в формате BMP и JPG.
3. **Дополнительные функции**
   1. изменение размера холста с помощью диалогового окна;
   2. команды автоматического расположения дочерних окон;
   3. диалоговое окно «О программе…».

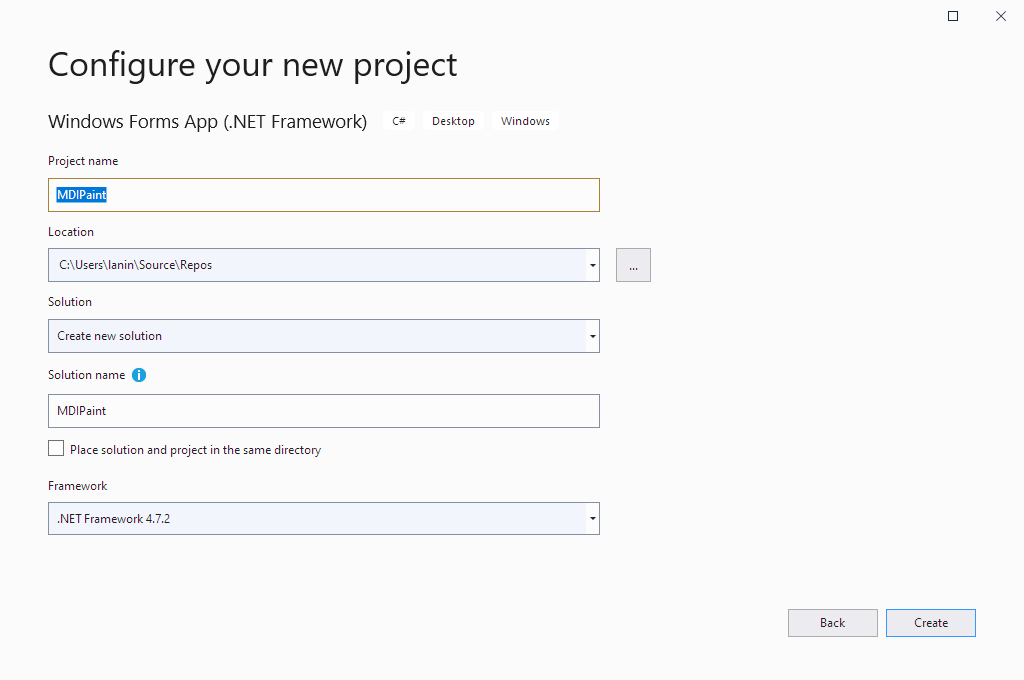
**Указания**

Создание и настройка проекта

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.

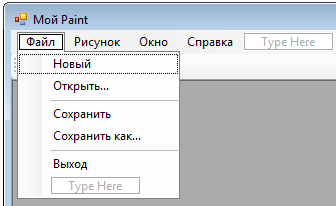


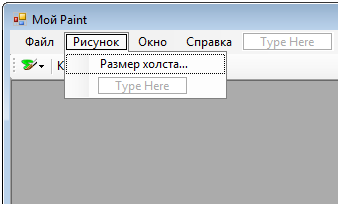
1. Для изменения свойств элементов экранных форм используется окно «Properties», которое отображается на экране при нажатии клавиши **F4** или выборе команды главного меню «View|Properties Window». Прежде чем изменить свойства какого-либо объекта его необходимо выделить щелчком мыши. Название выделенного объекта отображается в верхней части окна «Properties».

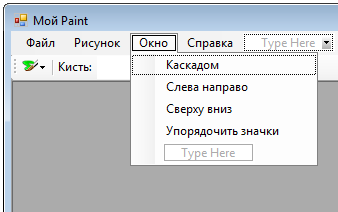
Установите значение свойства **Name** равным *MainForm* y созданной по умолчанию экранной формы проекта. Также установите свойство **IsMdiContainer** равным *True*, для того чтобы сделать форму родительской. Используя свойство **Text** формы, измените заголовок окна.

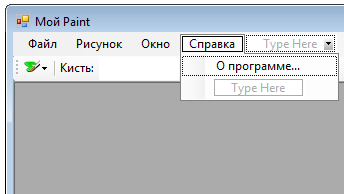
1. Отобразите на экране окно «Toolbox», если оно не видно на экране. Это можно сделать, нажав комбинацию клавиш «Ctrl+Alt+X» или выбрав соответствующий пункт меню «View». Элементы управления распределены по логическим группам, просмотреть все стандартные элементы управления можно в разделе «All Windows Forms».

Добавьте на форму элемент MenuStrip. Создайте структуру меню, как показано на рис. X

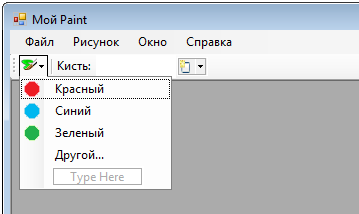




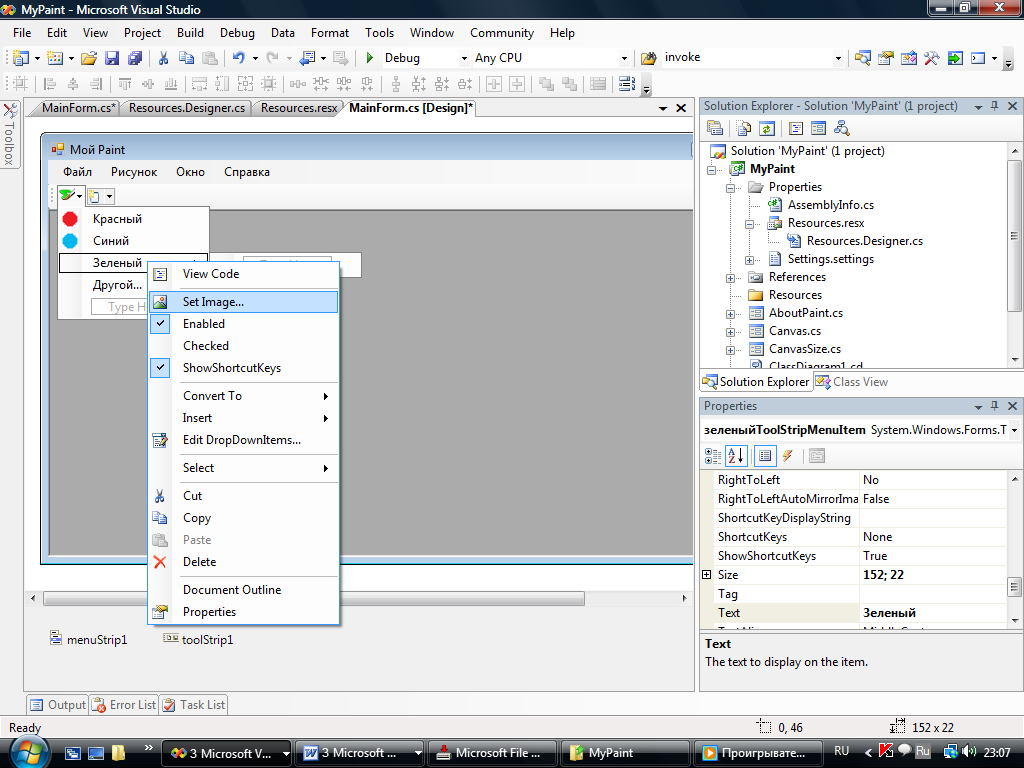


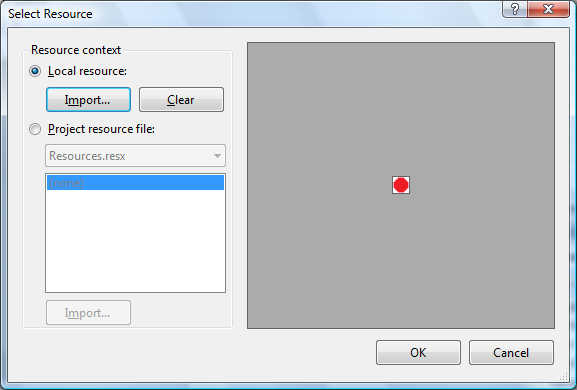


1. Выделите элемент управления MenuStrip и установите его свойство MdiWindowListItem равным *окноToolStripMenuItem* (имя пункта меню «Окно»). Использование данного свойства позволит автоматически отображать список дочерних окон в соответствующем пункте меню.
2. Добавьте элемент управления «ToolStrip», 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».

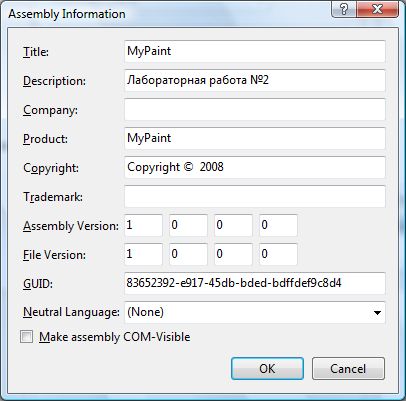
private void выходToolStripMenuItem\_Click(object sender, EventArgs e)

{

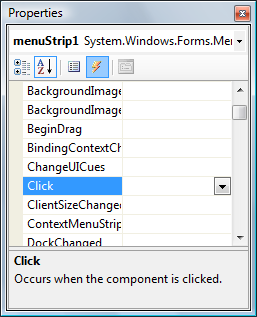
Application.Exit();

}

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

{

AboutPaint frmAbout = new AboutPaint();

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 *Canvas*.
2. Установите свойство BackColor формы *Canvas* равным AppWorkspace, свойство AutoScroll равным True.
3. На дочерней форме поместите элемент управления PictureBox. Для добавленного элемента установите свойство *Dock* в значение Fill.
4. Добавьте следующий программный код в дочернюю форму.

public partial class Canvas : Form

{

private int oldX, oldY;

public Canvas()

{

InitializeComponent();

}

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

{

if (e.Button == MouseButtons.Left)

{

Graphics g = pictureBox1.CreateGraphics();

g.DrawLine(new Pen(MainForm.CurColor, MainForm.CurWidth), oldX, oldY, e.X, e.Y);

oldX = e.X;

oldY = e.Y;

}

}

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

{

oldX = e.X;

oldY = e.Y;

}

}

1. В программный код главной формы добавьте объявление параметров рисования и обработчик команды открытия нового документа.

public partial class MainForm : Form

{

public static Color CurColor=Color.Black;

public static int CurWidth=3;

...

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

{

Canvas frmChild = new Canvas();

frmChild.MdiParent = this;

frmChild.Show();

}

}

1. Разработанное приложение пока обладает существенным недостатком: нарисованное изображение стирается при перерисовке формы. Для того чтобы избавиться от этого недостатка будем выполнять рисование не на форме, а на битовой карте (объекте Bitmap), отображая при перерисовке формы эту битовую карту на форме. Для реализации данного подхода объявите закрытый член класса bmp типа Bitmap, добавьте в конструктор строки создания объекта bmp, при рисовании объект класса Graphiсs необходимо получать из битовой карты. После рисования необходимо вызвать метод pictureBox1.Invalidate для отображения изменений.

public partial class Canvas : Form

{

private int oldX, oldY;

private Bitmap bmp;

public Canvas()

{

InitializeComponent();

bmp = new Bitmap(ClientSize.Width, ClientSize.Height);

pictureBox1.Image = bmp;

}

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

{

if (e.Button == MouseButtons.Left)

{

Graphics g = Graphics.FromImage(bmp);

g.DrawLine(new Pen(MainForm.CurColor, MainForm.CurWidth), oldX, oldY, e.X, e.Y);

oldX = e.X;

oldY = e.Y;

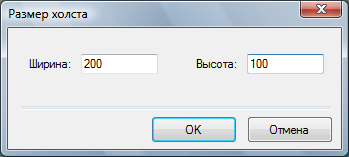
pictureBox1.Invalidate();

}

}

}

1. Добавьте в проект новую форму с именем CanvasSize. 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 рис. Х.



1. 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.
2. Добавьте в форму Canvas два метода для изменения высоты и ширины холста для рисования. При установке новых значений создается новая битовая карта, на которую выводится старое изображение.

public int CanvasWidth

{

get

{

return pictureBox1.Width;

}

set

{

pictureBox1.Width = value;

Bitmap tbmp = new Bitmap(value, pictureBox1.Width);

Graphics g = Graphics.FromImage(tbmp);

g.Clear(Color.White);

g.DrawImage(bmp, new Point(0, 0));

bmp = tbmp;

pictureBox1.Image = bmp;

}

}

public int CanvasHeight

{

get

{

return pictureBox1.Height;

}

set

{

pictureBox1.Height = value;

Bitmap tbmp = new Bitmap(pictureBox1.Width, value);

Graphics g = Graphics.FromImage(tbmp);

g.Clear(Color.White);

g.DrawImage(bmp, new Point(0,0));

bmp = tbmp;

pictureBox1.Image = bmp;

}

}

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…».

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

{

CanvasSize cs = new CanvasSize();

cs.CanvasWidth = ((Canvas)ActiveMdiChild).CanvasWidth;

cs.CanvasHeight = ((Canvas)ActiveMdiChild).CanvasHeight;

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

{

((Canvas)ActiveMdiChild).CanvasWidth = cs.CanvasWidth;

((Canvas)ActiveMdiChild).CanvasHeight = cs.CanvasHeight;

}

}

1. В главной форме добавьте возможность выбора цвета и размера кисти.

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

{

CurColor = Color.Red;

}

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

{

CurColor = Color.Blue;

}

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

{

CurColor = Color.Green;

}

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

{

ColorDialog cd=new ColorDialog();

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

CurColor = cd.Color;

}

private void txtBrushSize\_TextChanged(object sender, EventArgs e)

{

try

{

CurWidth=int.Parse(txtBrushSize.Text);

}

catch

{

MessageBox.Show("Значение должн быть целым числом.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

}

}

1. Добавьте в класс Canvas метод сохранения изображения, реализующий команду «Сохранить как…».

public void SaveAs()

{

SaveFileDialog dlg = new SaveFileDialog();

dlg.AddExtension = true;

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

ImageFormat[] ff ={ImageFormat.Bmp, ImageFormat.Jpeg};

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

{

bmp.Save(dlg.FileName, ff[dlg.FilterIndex - 1]);

}

}

1. Для реализации возможности открытия файла добавьте в класс Canvas конструктор.

public Canvas(String FileName)

{

InitializeComponent();

bmp = new Bitmap(FileName);

Graphics g = Graphics.FromImage(bmp);

pictureBox1.Width = bmp.Width;

pictureBox1.Height = bmp.Height;

pictureBox1.Image = bmp;

}

1. Реализуйте обработчики соответствующих команд главного меню.

private void открытьToolStripMenuItem2\_Click(object sender, EventArgs e)

{

OpenFileDialog dlg = new OpenFileDialog();

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

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

{

Canvas frmChild = new Canvas(dlg.FileName);

frmChild.MdiParent = this;

frmChild.Show();

}

}

private void сохранитьКакToolStripMenuItem2\_Click(object sender, EventArgs e)

{

((Canvas)ActiveMdiChild).SaveAs();

}

1. Самостоятельно реализуйте команду «Сохранить». При первом сохранении нового изображения запрашивается имя файла, при последующих имя файла не запрашивается.
2. Добавьте команды автоматического расположения дочерних окон.

LayoutMdi(MdiLayout.Cascade);

LayoutMdi(MdiLayout.ArrangeIcons);

LayoutMdi(MdiLayout.TileHorizontal);

LayoutMdi(MdiLayout.TileVertical);

1. Команды меню должны соответствовать состоянию программы. Например, если нет открытых окон, команды «Сохранить» и «Сохранить как…» должны быть неактивны.
2. При закрытии файла пользователю должно быть предложено сохранить измененный файл.