МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Российский государственный университет им. А.Н. Косыгина   
(Технологии. Дизайн. Искусство)»

Кафедра автоматизированных систем обработки информации и управления

Отчет по лабораторной работе № 10

по дисциплине «Технологии программирования»

Тема: «Создание многооконного приложения»

Выполнил: Стычинский С.В., МВИ-121

Проверили: Кузьмина Т. М. и Адаев Р. Б.

Москва 2023

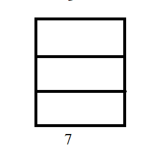
Цель работы

Научиться создавать многооконное приложение и инструменты для взаимодействия с ним.

Задание

Этапы выполнения работы.

1. Мой вариант(19 по списку) расположения дочерних форм:



Схемы заполнения форм:

* 1. На одну форму поместить рисунок, на остальных расположить отформатированный текст.
  2. На одну форму поместить отформатированный текст, на остальные формы поместить рисунки.
  3. На двух формах должны быть рисунки, на остальных (или остальной) должны располагаться тексты.
  4. На двух формах должны быть отформатированные тексты, на остальных (или остальной) должны располагаться рисунки.

1. Все рисунки и все текстовые документы должны сохраняться в файлах и из файлов выводится в нужную форму. Для выполнения этого задания нужно использовать компоненты openFileDialog, saveFileDialog.
2. Команды ввода, вывода, а также очистки компонента должны быть пунктами соответствующего меню.
3. На каждой дочерней форме должно присутствовать меню, но оно может быть главным или оперативным, вызываемым нажатием правой кнопки мыши.
   1. На одной форме должно быть главное меню, на остальных – оперативное.
   2. На одной форме должно быть оперативное меню, на остальных – главное.
4. Один рисунок должен быть редактируемым. На этом рисунке нужно позволить пользователю рисовать некоторую фигуру (можно эллипс или прямоугольник), фигура может быть фиксированной, но место положения ее должно меняться.
5. Все окна, содержащие текст должны иметь строку состояния, в которой должно быть записано имя документа (если документ был сохранен, то имя файла, в котором сохранен документ).
6. При наведении курсора мыши на окна, содержащие рисунки, должны появляться подсказки.

Распечатка программы

Класс FormMDI (Родительская форма)

public partial class FormMDI : Form

{

TextChangeForm f1 = new TextChangeForm();

EditablePicForm f2 = new EditablePicForm();

SecondPicForm f3 = new SecondPicForm();

public FormMDI()

{

InitializeComponent();

f1.MdiParent = this;

f1.Size = new Size(600, 300);

f1.StartPosition = FormStartPosition.Manual;

f1.Location = new Point(0, 0);

f1.Show();

f2.MdiParent = this;

f2.Size = new Size(600, 300);

f2.StartPosition = FormStartPosition.Manual;

f2.Location = new Point(0, 301);

f2.Show();

f3.MdiParent = this;

f3.Size = new Size(600, 300);

f3.StartPosition = FormStartPosition.Manual;

f3.Location = new Point(0, 601);

f3.Show();

}

}

Класс TextChangeForm

public partial class TextChangeForm : Form

{

public TextChangeForm()

{

InitializeComponent();

openFileDialog1.Filter = "Text files(\*.txt)|\*.txt|All files(\*.\*)|\*.\*";

saveFileDialog1.Filter = "Text files(\*.txt)|\*.txt|All files(\*.\*)|\*.\*";

toolStripStatusLabel1.Text = "Путь к файлу";

}

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

{

if (openFileDialog1.ShowDialog() == DialogResult.Cancel)

return;

string fileText = File.ReadAllText(openFileDialog1.FileName);

richTextBox1.Text = fileText;

toolStripStatusLabel1.Text = openFileDialog1.FileName;

}

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

{

if (saveFileDialog1.ShowDialog() == DialogResult.Cancel)

return;

string filename = saveFileDialog1.FileName;

File.WriteAllText(filename, richTextBox1.Text);

MessageBox.Show("Файл сохранен");

toolStripStatusLabel1.Text = saveFileDialog1.FileName;

}

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

{

richTextBox1.Clear();

toolStripStatusLabel1.Text = "Путь к файлу";

MessageBox.Show("Поле очищено");

}

}

Класс EditablePicForm

public partial class EditablePicForm : Form

{

Figure fig = new Figure();

public static Pen pen;

public static Bitmap bmp;

public static Graphics g;

public EditablePicForm()

{

InitializeComponent();

ToolTip t = new ToolTip(); //позволяет отображать всплывающую подсказку при наведении курсора

t.SetToolTip(pictureBox1, "Нажмите ПКМ для взаимодействия"); //подсказка

pictureBox1.ContextMenuStrip = contextMenuStrip1; //определяем границы для контексного меню

toolStripStatusLabel1.Text = "Путь к файлу"; //нижняя строка имени файла

}

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

{

if (openFileDialog1.ShowDialog() == DialogResult.Cancel)

return;

pictureBox1.SizeMode = PictureBoxSizeMode.Zoom; //Масштабирует изображение, чтобы оно соответствовало размеру PictureBox и при этом сохранял свои пропорции.

pictureBox1.Image = Image.FromFile(openFileDialog1.FileName);

toolStripStatusLabel1.Text = openFileDialog1.FileName; //в статус записывается имя файла

}

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

{

if (saveFileDialog1.ShowDialog() == DialogResult.Cancel)

return;

if (pictureBox1.Image == null)

{

MessageBox.Show("Невозможно сохранить пустое изображение");

return;

}

else

{

Bitmap bmp = new Bitmap(pictureBox1.Image);

//при сохранении файла бывали ошибки, поэтому обернуто в try catch

try

{

bmp.Save(saveFileDialog1.FileName, ImageFormat.Jpeg);

MessageBox.Show("Файл сохранен");

}

catch (Exception ex)

{

MessageBox.Show("Ошибка сохранения");

}

}

}

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

{

if (pictureBox1.Image != null)

{

fig.ShowDialog();

Graphics g = Graphics.FromImage(pictureBox1.Image);

Pen pen = new Pen(Color.Red, 5);

g.DrawRectangle(pen, Figure.x, Figure.y, 50, 50); //рисует прямоугольник размерами 50х50 по заданным координатам в форме

pictureBox1.Refresh();

}

else

{

MessageBox.Show("Изображение отсутстввует");

}

}

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

{

pictureBox1.Image = null;

toolStripStatusLabel1.Text = "Путь к файлу";

MessageBox.Show("Изображение удалено");

}

}

Класс SecondPicForm

public partial class SecondPicForm : Form

{

public static Pen pen;

public static Bitmap bmp;

public static Graphics g;

public SecondPicForm()

{

InitializeComponent();

ToolTip t = new ToolTip(); //позволяет отображать всплывающую подсказку при наведении курсора

t.SetToolTip(pictureBox1, "Нажмите ПКМ для взаимодействия"); //подсказка

pictureBox1.ContextMenuStrip = contextMenuStrip1; //определяем границы для контексного меню

toolStripStatusLabel1.Text = "Путь к файлу"; //нижняя строка имени файла

}

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

{

if (openFileDialog1.ShowDialog() == DialogResult.Cancel)

return;

pictureBox1.SizeMode = PictureBoxSizeMode.Zoom; //Масштабирует изображение, чтобы оно соответствовало размеру PictureBox и при этом сохранял свои пропорции.

pictureBox1.Image = Image.FromFile(openFileDialog1.FileName);

toolStripStatusLabel1.Text = openFileDialog1.FileName; //в статус записывается имя файла

}

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

{

if (saveFileDialog1.ShowDialog() == DialogResult.Cancel)

return;

if (pictureBox1.Image == null)

{

MessageBox.Show("Невозможно сохранить пустое изображение");

return;

}

else

{

Bitmap bmp = new Bitmap(pictureBox1.Image);

//при сохранении файла бывали ошибки, поэтому обернуто в try catch

try

{

bmp.Save(saveFileDialog1.FileName, ImageFormat.Jpeg);

MessageBox.Show("Файл сохранен");

}

catch (Exception ex)

{

MessageBox.Show("Ошибка сохранения");

}

}

}

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

{

pictureBox1.Image = null;

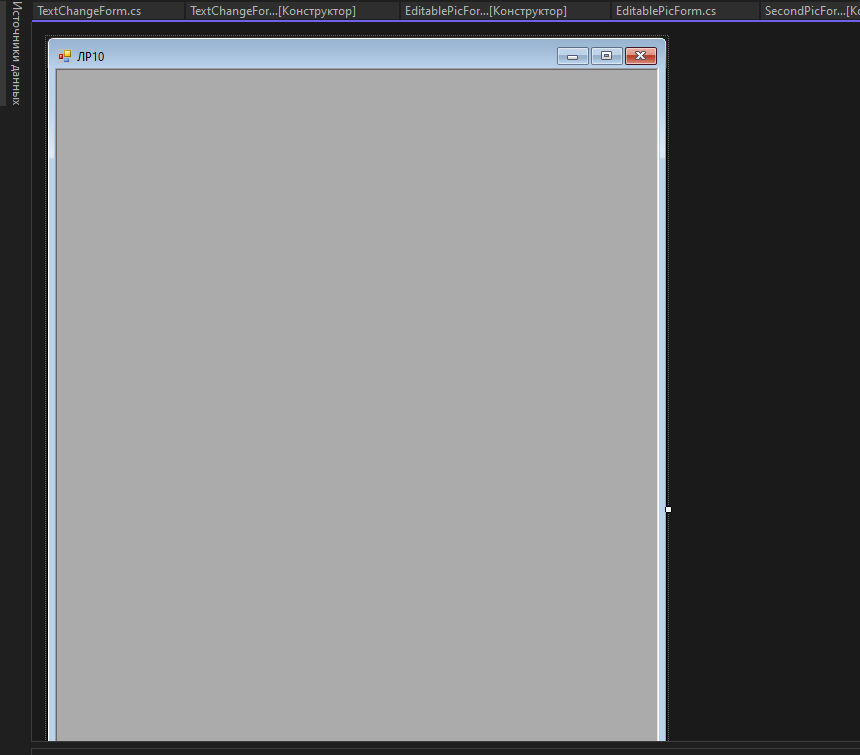
toolStripStatusLabel1.Text = "Путь к файлу";

MessageBox.Show("Изображение удалено");

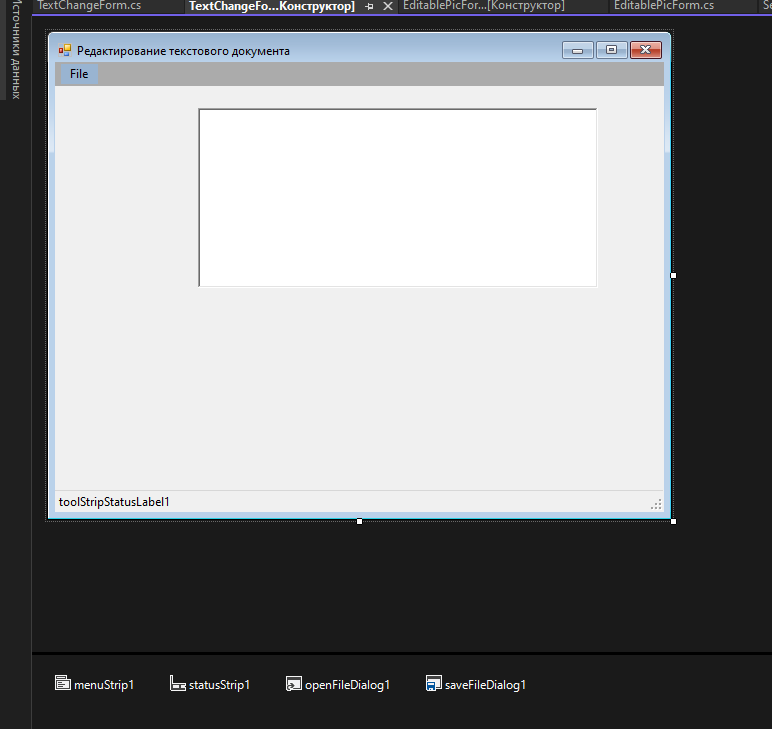
}

}

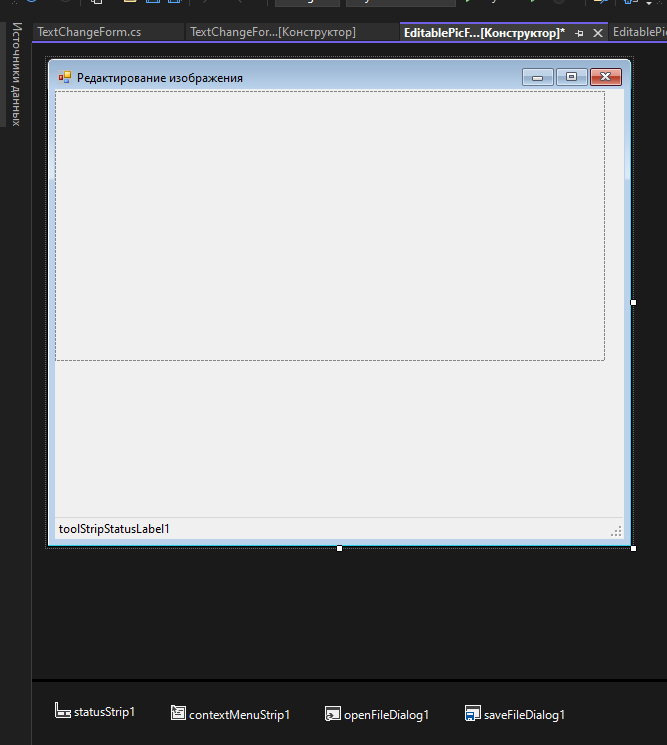
Форма FormMDI



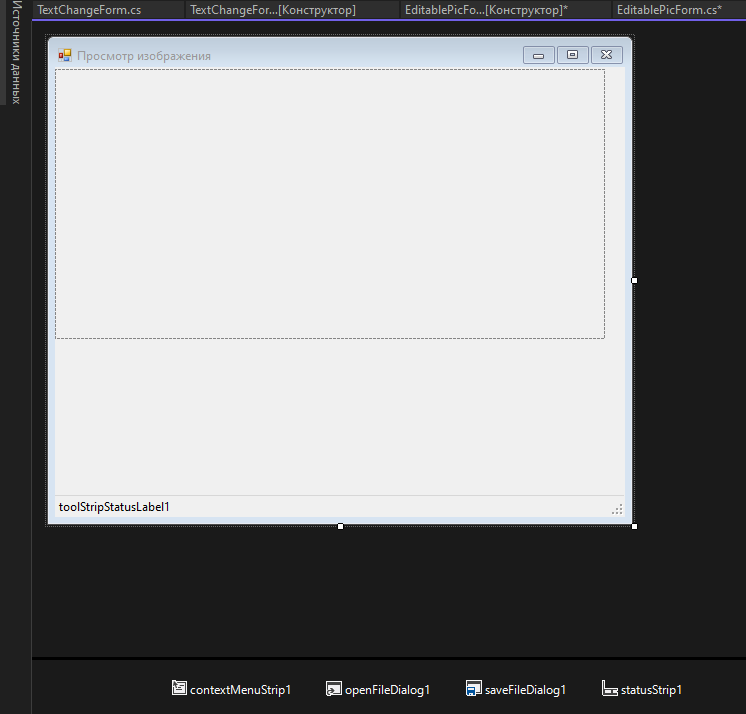
Форма TextChangeForm



Форма EditablePicForm



Форма SecondPicForm



Результат работы программы

