**Лабораторная работа № 6**

**Широков Ф. А. з22928/2**

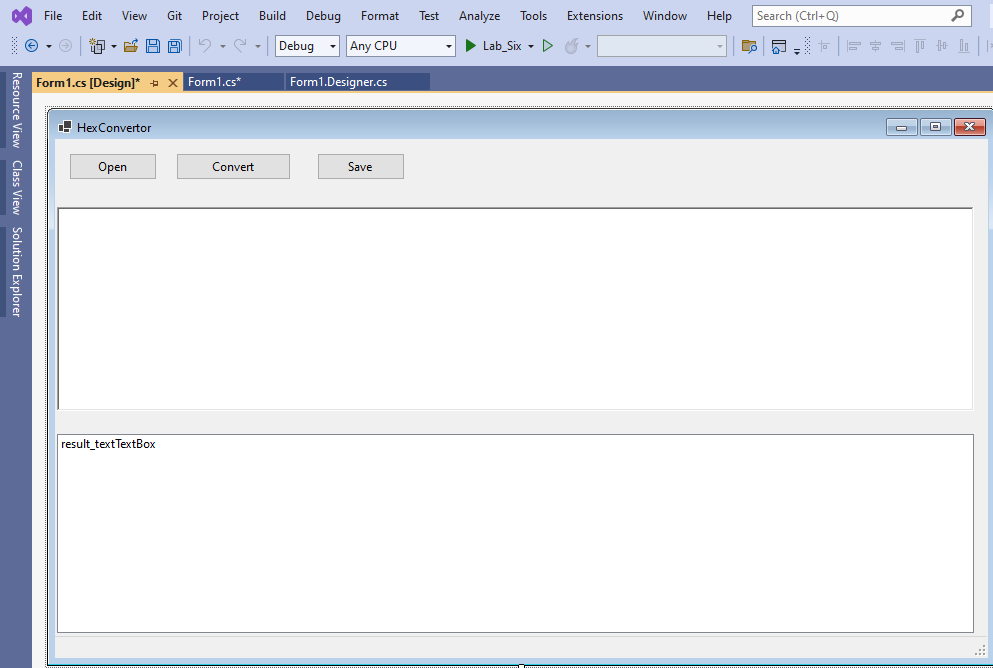
**Тема:** Создание проектов обработки текстовых файлов

**Вариант 10**: В текстовом файле хранятся шестнадцатеричные числа (каждое число на отдельной строке). Разработать программу, которая загружает в текстовый редактор файл по выбору пользователя и выполняет перевод этих чисел в десятичную систему счисления. Если какая-либо строка в файле содержит не шестнадцатеричное число (то есть в ней встречаются символы, отличные от нужного набора), то в качестве результата вывести прочерк. Результаты перевода сохранить в другой файл.

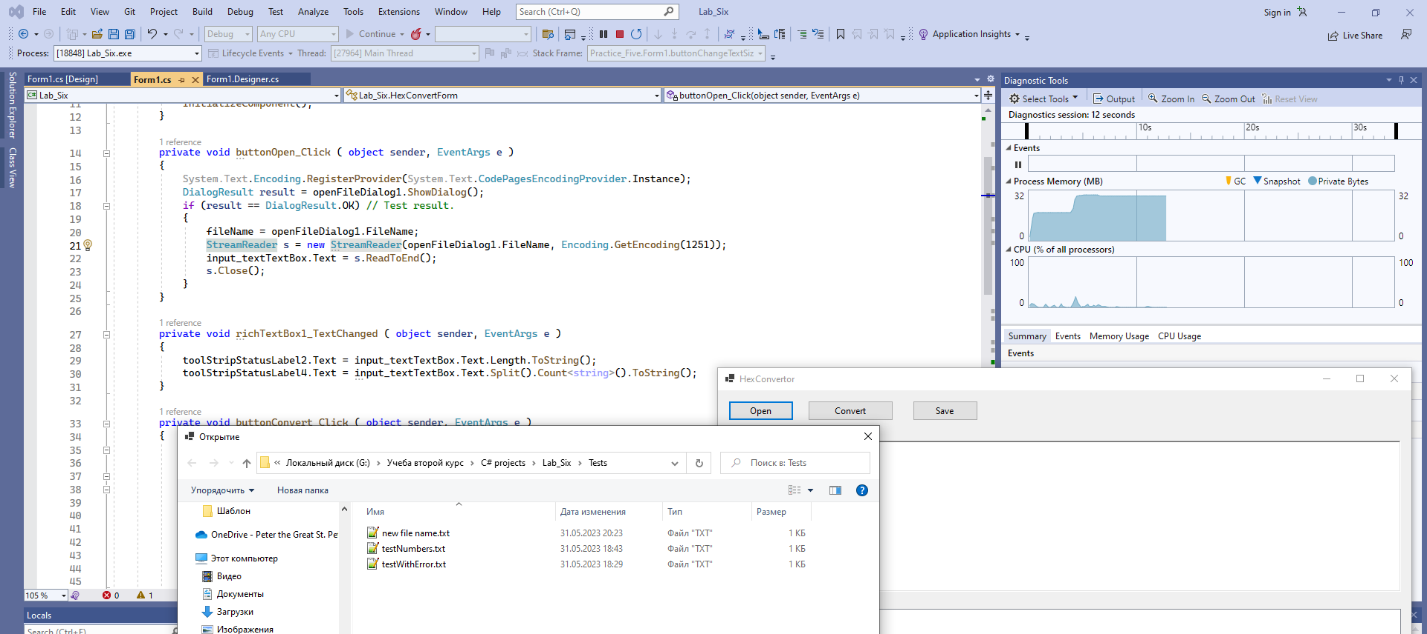
**Цель работы:** получить практические навыки разработки проектов с использованием компонента текстовый редактор для чтения и обработки текстового файла

**Ход выполнения:**

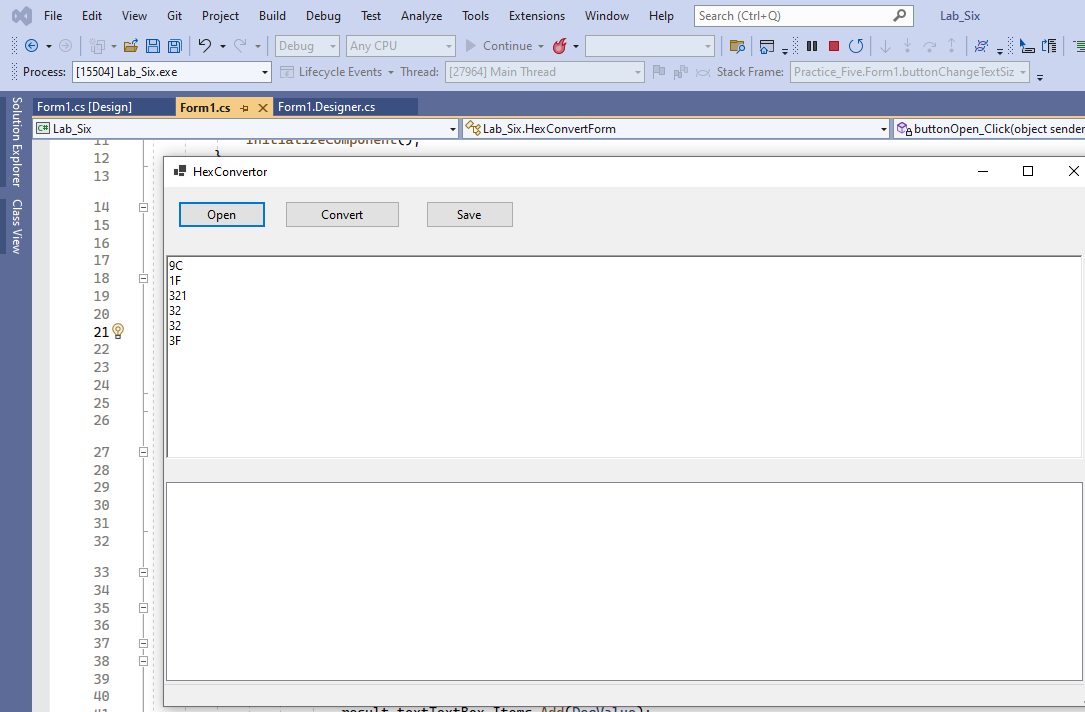
1. Создадим дизайн формы учитывая тех. задание. Программа должна иметь возможность открыть файл, конвертировать его по определенному алгоритму и сохранять.

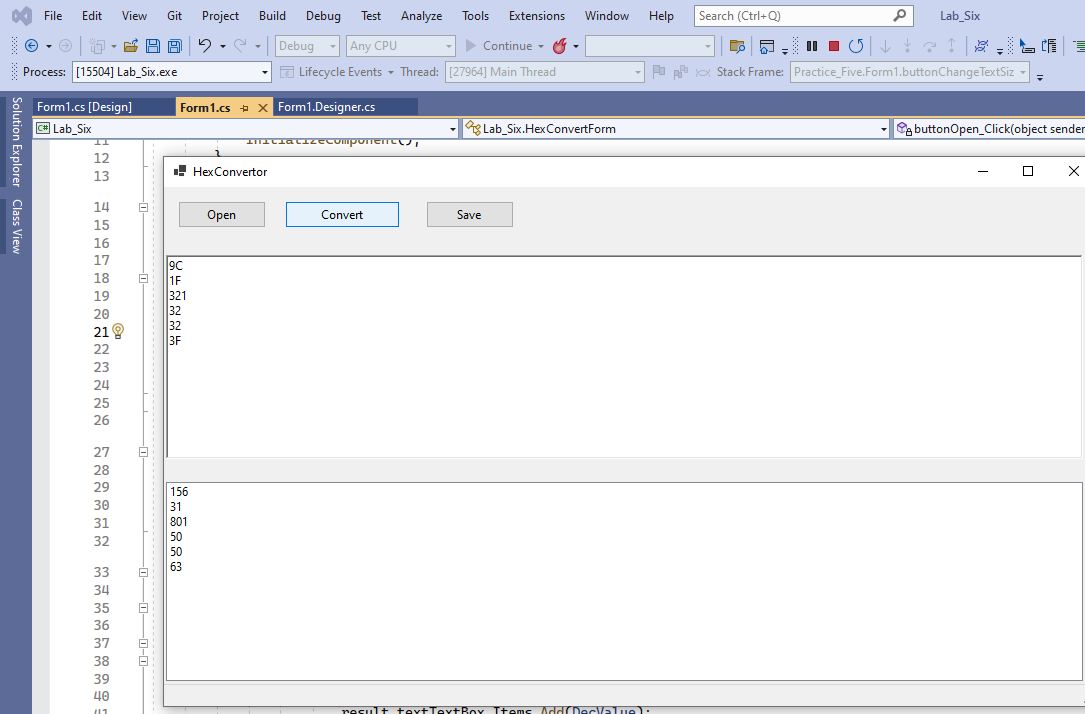


2. Реализуем метод для открытия файла. Для этого мы будем использовать объект DialogResult. После нажатия кнопки происходить открытия диалогового окна с возможностью выбора файла. После выбора файла начинаем записывать данные с помощью StreamReader в TextBox нашей программы.



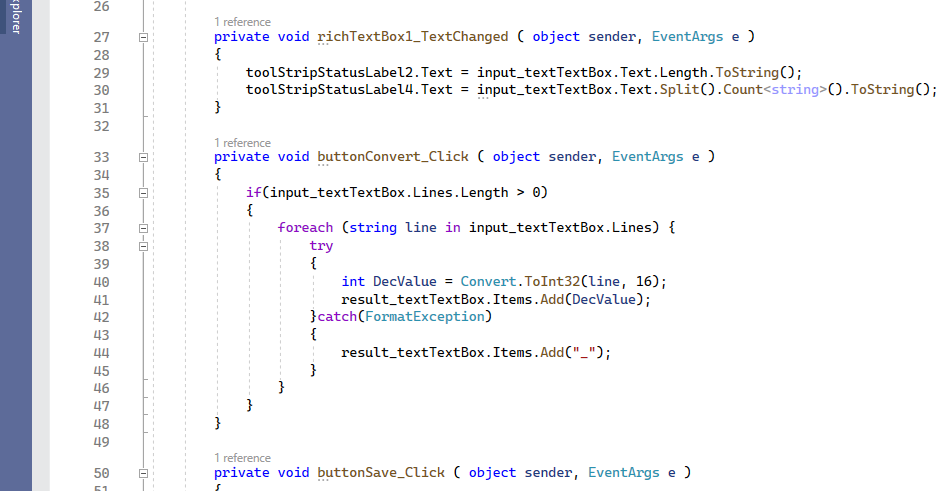
3. Далее, для запуска процесса конвертирования нужно нажать на кнопку Convert.



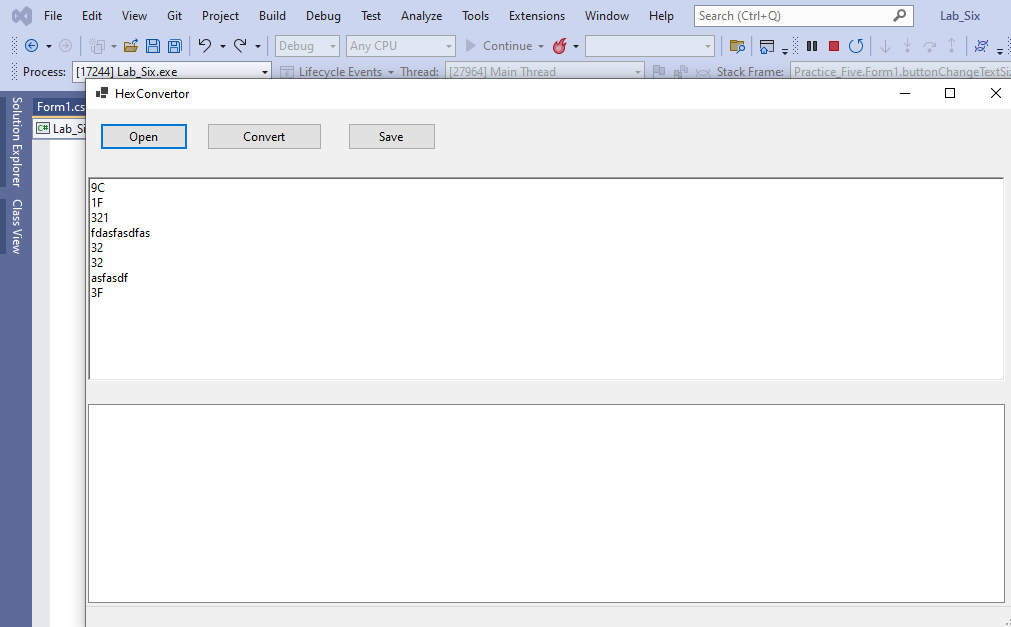


4. Так как по условию задания возможно, что вместо корректных данных будут переданы

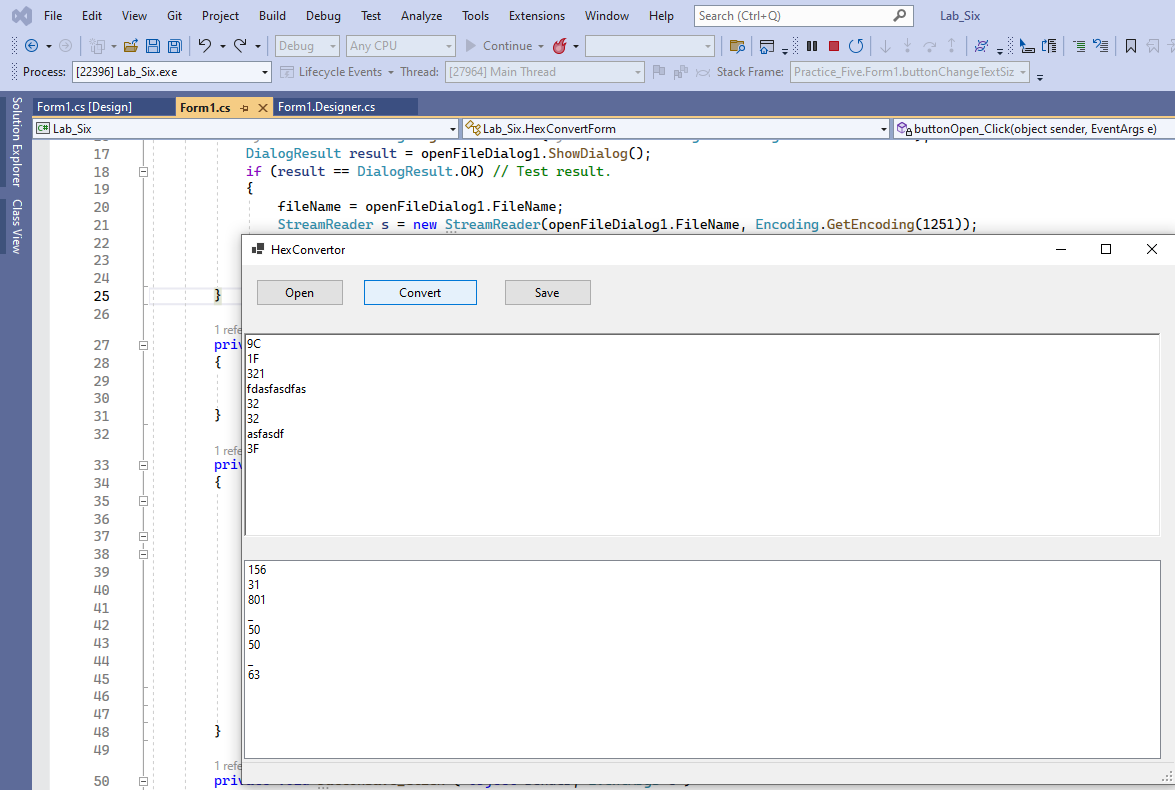
не числовые значения, в методе Convert предусмотрена обработка исключений и реализована вставка символа нижнего подчеркивания (‘\_’) вместо некорректного символа.



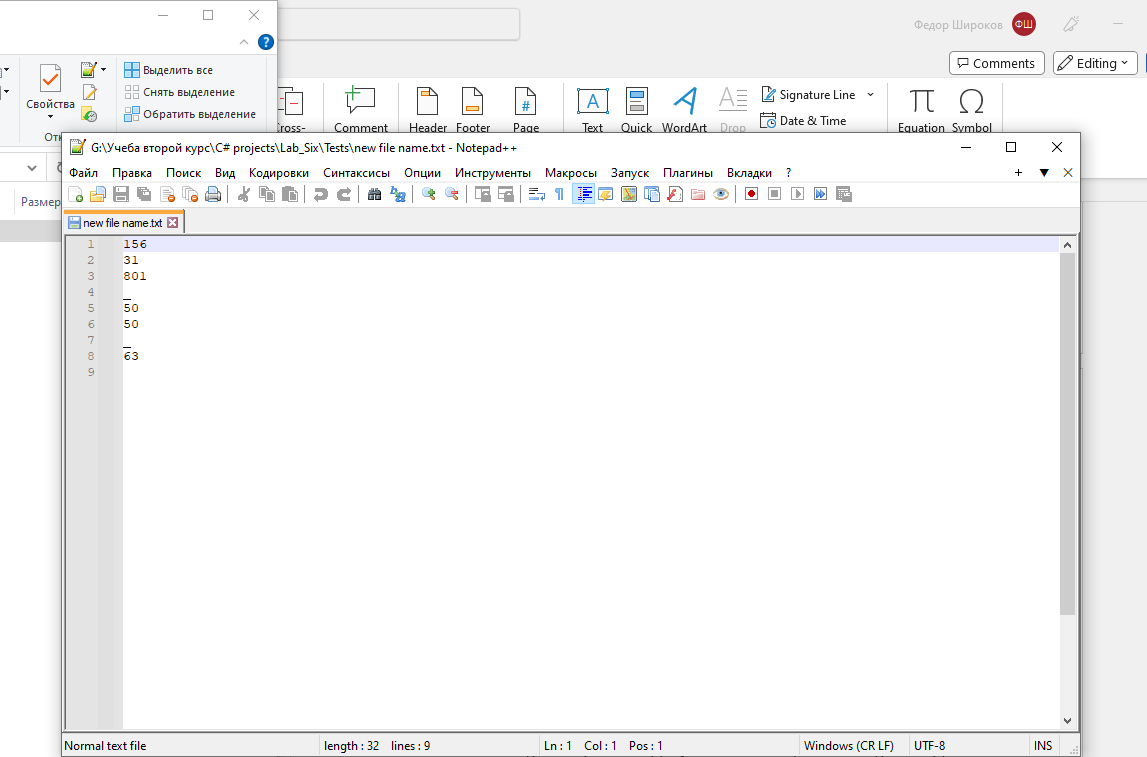
5. Проверим обработку исключения, подадим файл со строковыми значениями.



После нажатия на Convert.



6. Теперь осталось проверить, что в файл записываются корректные данные.



**Код программы:**

using System.Text;

namespace Lab\_Six

{

public partial class HexConvertForm : Form

{

string fileName = "";

public HexConvertForm ( )

{

InitializeComponent();

}

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

{

System.Text.Encoding.RegisterProvider(System.Text.CodePagesEncodingProvider.Instance);

DialogResult result = openFileDialog1.ShowDialog();

if (result == DialogResult.OK) // Test result.

{

fileName = openFileDialog1.FileName;

StreamReader s = new StreamReader(openFileDialog1.FileName, Encoding.GetEncoding(1251));

input\_textTextBox.Text = s.ReadToEnd();

s.Close();

}

}

private void richTextBox1\_TextChanged ( object sender, EventArgs e )

{

toolStripStatusLabel2.Text = input\_textTextBox.Text.Length.ToString();

toolStripStatusLabel4.Text = input\_textTextBox.Text.Split().Count<string>().ToString();

}

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

{

if(input\_textTextBox.Lines.Length > 0)

{

foreach (string line in input\_textTextBox.Lines) {

try

{

int DecValue = Convert.ToInt32(line, 16);

result\_textTextBox.Items.Add(DecValue);

}catch(FormatException)

{

result\_textTextBox.Items.Add("\_");

}

}

}

}

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

{

SaveFileDialog sfd = new SaveFileDialog();

sfd.Filter = "Text File|\*.txt";

sfd.FileName = "new file name";

sfd.Title = "Save Text File";

if (sfd.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

using (StreamWriter bw = new StreamWriter(File.Create(sfd.FileName)))

{

List<string> lines = new List<string>();

foreach (var line in result\_textTextBox.Items)

{

lines.Add(line.ToString());

}

foreach (string addedline in lines)

{

bw.WriteLine(addedline);

}

bw.Dispose();

}

}

}

}

}