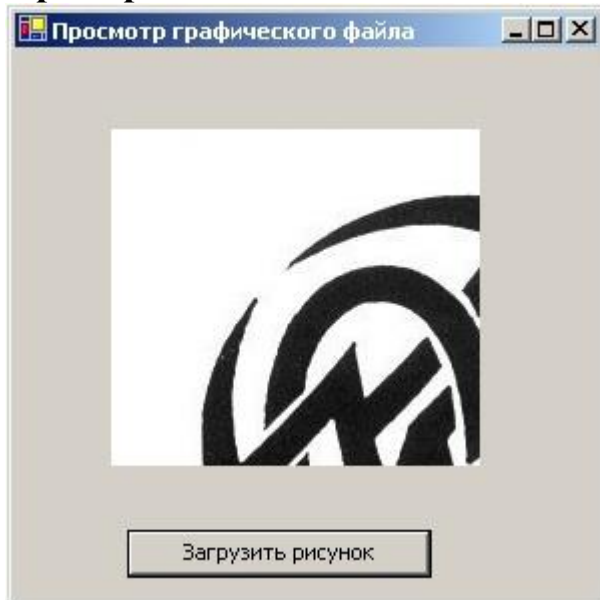


Задание №1

Выполнить: Разработайте приложение, которое выводит на экран выбранный графический файл.

Пример выполнения:



объект

свойство
name

форма

контейнер для изображения (PictureBox) pct

кнопка (button)

btn

диалоговое окно (OpenFileDialog)

file1

1. Создайте новый проект и расположите элементы управления *PictureBox* (pct) и *Button* (btn) на новой форме, как показано на рисунке. Добавьте также элемент управления *OpenFileDialog*. Это скрытый элемент, не отображается в дизайне формы. Для свойства *name* элемента задайте имя *file1*.
2. Предположим, необходимо открывать только графические файлы формата *jpg*. В таком случае, необходимо задать фильтр для файлов. Сделать это следует при загрузке формы, т.е. нужно запрограммировать событие *Load* для формы (дважды щелкнув по форме в конце дизайна формы):

```
private void Form1_Load(object sender, EventArgs e)
{
    file1.Filter = "(*.jpg)|*.jpg";
}
```

До прямого слэша указывается, как будет отображаться фильтр в окне, а после – сама маска фильтра.

3. Теперь осталось вывести изображение из графического файла в *PictureBox* (*pct*). Для этого запрограммируем событие *click* для кнопки:

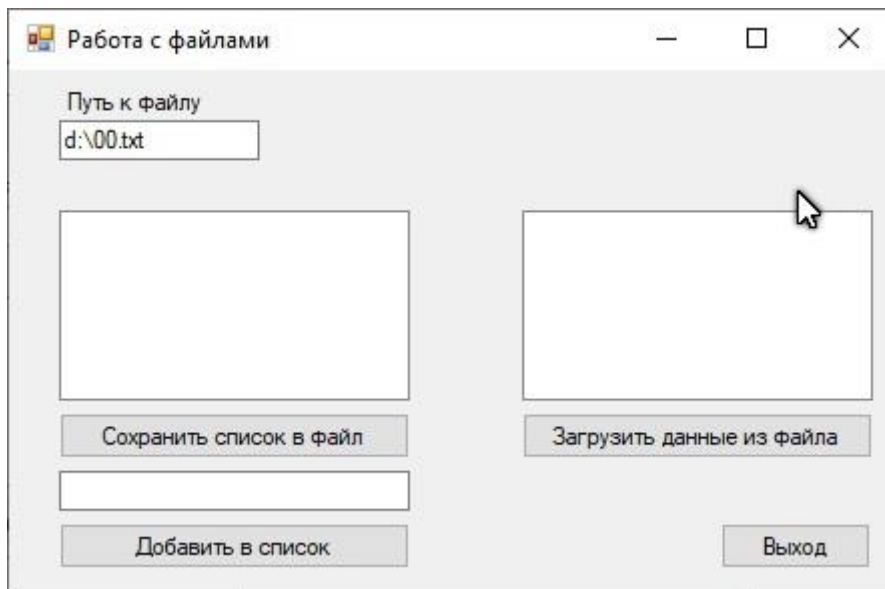
```
// создаем переменную fname строкового типа
string fname;
// открываем Проводник
file1.ShowDialog();
// используем переменную для хранения имени выбранного файла
fname = file1.FileName;
// загружаем файл в элемент PictureBox
pct.Image = Image.FromFile(fname);
```

4. Протестируйте результат.
5. Добавьте на форму текстовое окно для вывода полного имени открываемого графического файла.
6. Добавьте на форму элемент управления *SaveFileDialog* – диалоговое окно для сохранения файла и кнопку *Сохранить файл*. Необходимо использовать метод *ShowDialog*, СВОЙСТВО *file1.SafeFileName* И *pct.Image.Save=....*

Задание №2

Выполнить: Разработайте приложение, которое сохраняет в файл информацию из списка (*ListBox*) и загружает из файла информацию обратно в список.

Пример выполнения:



[Название проекта: Lab7_2, название файла Lab7_2.cs]

1. Поместите на форму элементы:
список *ListBox (lst)*, *ListBox (lstFromFile)* текстовое поле *TextBox (txt)*,
кнопки: *btnAdd* («Добавить элемент в список»), *btnSave* («Сохранить
в файле»), *btnOpen* («Загрузить из файла»). Расположите правильно
элементы (рисунок).

2. Добавьте пространство имен для работы с файлами:

```
using System.IO;
```

3. При нажатии на *btnAdd* («Добавить элемент в список») в
список *lst* должна добавиться строка, записанная в текстовое
окно *txt*, а само окно при этом должно очиститься. Добавьте код для
события *click* кнопки *btnAdd*:

```
private void btnAdd_Click(object sender, EventArgs e)
{
    lst.Items.Add(txt.Text);
    txt.Clear();
}
```

4. По щелчку на кнопке *btnSave* вся информация со списка должна
добавляться во введенный пользователем в текстовое окно
текстовый файл. Для этого добавьте процедуру для
события *Click* кнопки:

```

private void btnSave_Click(object sender, EventArgs e)
{
    string fileName = txtFileName.Text; // путь к файлу
    if (File.Exists(fileName))
    {
        File.Delete(fileName); // если файл существует - удаляем его
    }
    using (FileStream fs = File.Create(fileName, 1024)) // класс для работы с файлами
    // класс для работы с данными файла в двоичной виде
    using (BinaryWriter bw = new BinaryWriter(fs))
    {
        for (var i = 0; i < lst.Items.Count; i++) // пока не конец списка
        {
            bw.Write(lst.Items[i].ToString()); // записываем в файл каждый пункт
        }
        bw.Close();
        fs.Close();
    }
}

```

5. Событие *Click* кнопки *btnOpen* («Загрузить из файла») запрограммируйте таким образом, чтобы в список помещались все строки из файла:

```

private void btnOpen_Click(object sender, EventArgs e)
{
    string fileName = txtFileName.Text;
    lstFromFile.Items.Clear();
    using (FileStream fs = new FileStream(fileName, FileMode.Open))
    using (BinaryReader br = new BinaryReader(fs))
    {
        // метод PeekChar() возвращает следующий прочитанный символ
        // если символов нет - возвращается -1
        while (br.PeekChar() != -1)
        {
            // добавляем в список очередную прочитанную строку
            lstFromFile.Items.Add(br.ReadString());
        }
        br.Close();
        fs.Close();
    }
}

```

6. Запустите приложение, внесите в список *lst* несколько строк и попытайтесь сохранить их в файл (желательно с расширением *.txt*: например, [*proba.txt*](#)). Если не написан путь к файлу, то файл сохранится в папке */bin/Debug* Вашего проекта.