

Лабораторная работа 4. Использование окон диалога в формах

Цель работы

Изучение способов использования компонентов, представляющие диалоговые окна и получение навыков по работе с окнами диалога.

Упражнение 1. Использование компонента `SaveFileDialog`

Чтобы пользователи могли сохранять файлы, можно использовать встроенный компонент **`SaveFileDialog`**.

В этом упражнении Вы отобразите диалоговое окно, используя метод **ShowDialog**. Затем с помощью поля **DialogResult.OK** проверите, нажал ли пользователь кнопку **OK**.

Для реализации отображения диалогового окна обозревателя папок выполните:

1. Создайте приложение Windows Forms, укажите имя **TestStandartDialog**.

2. Добавьте элемент **MenuItem**, задайте имя первого пункта меню **Файл** и команду **Сохранить как...**

3. Добавьте в форму элемент управления **richTextBox**, оставив имя по умолчанию **richTextBox1**. Свойству **Dock** установите **Fill**.

4. Добавьте в форму компонент **SaveFileDialog**. Проверьте, что в области компонентов появился компонент **saveFileDialog1**.

5. Дважды щелкните кнопку, чтобы добавить в редактор кода обработчик событий по умолчанию.

6. В обработчике событий добавьте следующий код для отображения диалогового окна **Сохранение файла**. Этот код сохраняет текст, введенный в элемент управления **richTextBox**, в текстовый файл в указанной папке.

```
saveFileDialog1.Filter = "txt files (*.txt)|*.txt";
if(saveFileDialog1.ShowDialog() ==
System.Windows.Forms.DialogResult.OK
&& saveFileDialog1.FileName.Length > 0)
{
    richTextBox1.SaveFile(saveFileDialog1.FileName,
        RichTextBoxStreamType.PlainText);
}
```

7. Постройте и протестируйте приложение.

8. В открывшейся форме введите какой-либо текст в текстовое поле.

9. Выберите команду **Сохранить как...** и сохраните файл (имя и место для сохранения файла выберите по своему усмотрению).

10. Убедитесь, что текстовый файл находится в указанном месте.

Упражнение 2. Использование компонента ColorDialog

Для отображения диалогового окна цветовой палитры можно использовать встроенный компонент **ColorDialog** вместо того, чтобы создавать свое собственное диалоговое окно.

В этом упражнении Вы дополните приложение **TestStandartDialog**, чтобы дать пользователям возможность выбрать цвет и применить его в форме Windows после указания соответствующей команды меню.

Для реализации отображения диалогового окна цветовой палитры выполните:

1. Для элемента **MenuItem** задайте имя второго пункта меню – **Формат** и команду – **Цвет фона**.

2. Добавьте в форму компонент **ColorDialog**.

3. Проверьте, что в области компонентов появился компонент **colorDialog1**.

4. Дважды щелкните кнопку **Цвет фона**, чтобы создать обработчик событий по умолчанию в редакторе кода.

5. В обработчике событий добавьте следующий код для отображения диалогового окна выбора цвета и изменения фонового цвета в соответствии с выбором пользователя:

```
if (colorDialog1.ShowDialog() == DialogResult.OK)
{
    richTextBox1.BackColor = colorDialog1.Color;
}
```

6. Постройте и протестируйте приложение.

Упражнение 3. Использование компонента FontDialog

Для отображения диалогового окна выбора шрифтов можно использовать встроенный компонент **FontDialog** вместо того, чтобы создавать свое собственное диалоговое окно.

В этом упражнении Вы дополните приложение **TestStandartDialog**, чтобы дать пользователям возможность выбрать шрифт в диалоговом окне и затем применить его к тексту.

Для реализации отображения диалогового окна выбора шрифтов выполните:

1. Задайте в меню **Формат** новую команду – **Шрифт**.
2. Перетащите в форму компонент **FontDialog**.
3. Проверьте, что в области компонентов появится компонент **fontDialog1**.

4. Дважды щелкните команду **Шрифт**, чтобы создать в редакторе кода обработчик событий по умолчанию.

5. В обработчик событий добавьте следующий код для отображения диалогового окна выбора шрифта текста в окне и изменения шрифта текста в соответствии с выбором пользователя:

```
if (fontDialog1.ShowDialog() == DialogResult.OK)
{
    richTextBox1.Font = fontDialog1.Font;
}
```

6. Постройте и протестируйте приложение.

Упражнение 4. Использование компонента OpenFileDialog

Чтобы пользователи могли выбрать текстовый файл и загрузить его в элемент управления **RichTextBox** в форме Windows Forms, можно использовать компонент **OpenFileDialog**.

В этом упражнении Вы дополните приложение **TestStandartDialog**, чтобы дать пользователям возможность открыть текстовый файл.

1. Задайте в меню **Файл** новую команду **Открыть...**

2. Перетащите в форму компонент **OpenFileDialog**.

В области компонентов появился компонент **openFileDialog1**.

3. Дважды щелкните команду **Открыть...**, чтобы создать в редакторе кода обработчик событий по умолчанию.

4. В обработчик событий добавьте следующий код для отображения диалогового окна открытия файла:

```
Stream myStream = null;
OpenFileDialog openFileDialog1 = new OpenFileDialog();
openFileDialog1.InitialDirectory = @"c:\";
openFileDialog1.Filter = "txt files (*.txt)|*.txt|All
files (*.*)|*.*";
openFileDialog1.FilterIndex = 2;
if (openFileDialog1.ShowDialog() == DialogResult.OK)
{
    try
    {
        if ((myStream = openFileDialog1.OpenFile()) != null)
        {
            using (myStream)
            {
                richTextBox1.LoadFile(openFileDialog1.FileName,
                    RichTextBoxStreamType.PlainText);
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error: Could not read file from disk: "
            + ex.Message);
    }
}
```

5. Постройте и протестируйте приложение.