

Лабораторная работа №13

Изучение процесса работы с диалоговыми окнами в приложениях WPF

1Цель работы

1.1 Изучить процесс создания и применения стандартных и пользовательских диалоговых окон в приложениях WPF.

2Литература

2.1 <https://metanit.com/sharp/wpf/> - гл.20 - Окна

3Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4Основное оборудование

4.1 Персональный компьютер.

5Задание

5.1 Использование диалогового окна открытия файла

5.1.1 Создать WPF-приложение.

5.1.2 Добавить в приложение Image (элемент отображения изображений) и кнопку «Открыть».

При нажатии на кнопку «Открыть» предоставить пользователю возможность выбора изображений с расширением png, jpg (jpeg), bmp.

Для открытия использовать OpenFileDialog (требуется подключение пространства имен Microsoft.Win32).

У диалогового окна настроить фильтрацию.

Если ShowDialog() у диалогового окна вернул true, то загружать выбранное изображение в Image и изменять заголовок формы на название выбранного файла.

Для загрузки изображения в Image используется следующий код:

```
контрол.Source = new BitmapImage(new Uri(dialog.FileName));
```

5.2 Использование диалогового окна сохранения файла

5.2.1 Создать WPF-приложение.

5.2.2 Добавить в приложение многострочное поле ввода и кнопку «Сохранить».

При нажатии на кнопку «Сохранить» предоставить пользователю возможность указания имени текстового файла (допустимые расширения: txt, cs, html, css, js, sql).

Для сохранения использовать SaveFileDialog (требуется подключение пространства имен Microsoft.Win32).

У диалогового окна настроить фильтрацию.

Если ShowDialog() у диалогового окна вернул true, то записывать данные из поля ввода в указанный файл.

5.3 Настройка MessageBox

5.3.1 Добавить в приложение из п.5.1 или п.5.2 обработчик события окна Closing.

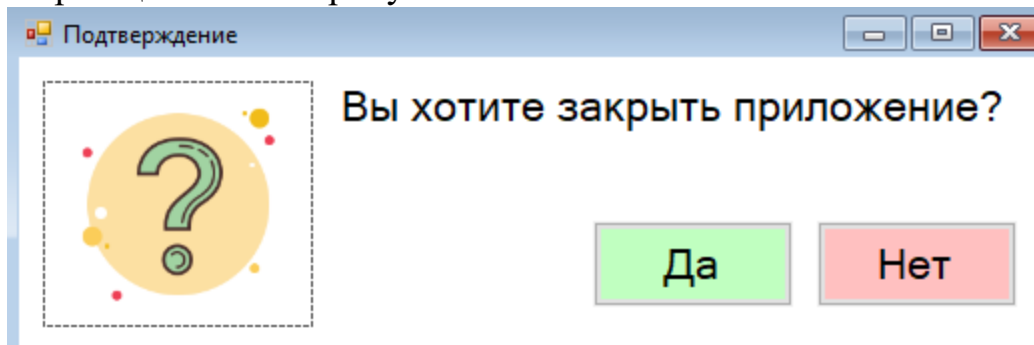
5.3.2 В обработчике события Closing создать MessageBox, в котором настроить текст сообщения «Вы хотите закрыть приложение?», заголовок «Подтверждение», иконку с вопросительным знаком и кнопки Да/Нет.

5.3.3 Если пользователь нажал Нет в MessageBox, то для отмены закрытия написать:
`e.Cancel = true;`

5.4 Создание диалогового окна для подтверждения закрытия окна

Создать свой аналог MessageBox из п.5.3 и использовать его в том окне, в котором не использован MessageBox.

Образец показан на рисунке:



Для выполнения при нажатии Нет изменять значение DialogResult окна (например, на значение true или значение типа DialogResult) и обрабатывать это значение в вызывающей форме.

5.5 Создание диалогового окна для выбора цвета фона окна

5.5.1 Создать WPF-приложение.

5.5.2 Добавить в приложение новое окно, на котором разместить кнопку «ОК» и три слайдера для того, чтобы пользователь мог указать число от 0 до 255 для цветовых составляющих (Red, Green, Blue). Снабдить слайдеры метками, чтобы пользователь понимал, что он должен указать. При нажатии на ОК изменять значение DialogResult окна на true.

5.5.3 Добавить в окно со слайдерами открытое свойство на чтение, возвращающее цвет на основе данных слайдеров:

`Color.FromRgb(красный, зеленый, синий)`

5.5.4 Реализовать открытие окна со слайдерами с главного окна. Если пользователь нажал в окне выбора цвет ОК, то изменять цвет фона главного окна.

Для указания цвета в главном окне используется следующий код:

`Background = new SolidColorBrush(цвет);`

6 Порядок выполнения работы

6.1 Выполнить все задания из п.5 в одном решении LabWork13. Каждый проект – приложение WPF.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое диалоговое окно?

8.2 Что такое OpenFileDialog и какое его основное свойство?

- 8.3 Что такое `SaveFileDialog` и какое его основное свойство?
- 8.4 Что такое `MessageBox` и какие настройки можно ему задать?
- 8.5 Как сделать доступными данные пользовательского диалогового окна вызывающим его окнам?
- 8.6 Какие значения может принимать переменная `DialogResult`?