

Лабораторная работа №36

Разработка приложения с несколькими формами

1 Цель работы

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

2 Литература

2.1 <https://metanit.com/sharp/wpf/> – гл.19 Окона

3 Подготовка к работе

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

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Создание недиалоговых окон:

5.1.1 Создать WPF-приложение.

5.1.2 Добавить в приложение:

- пункт меню «Файл» с подпунктом «Новое окно»,

- поле ввода с разрешением многострочного ввода, занимающее весь экран.

5.1.3 При нажатии на «Новое окно» создавать новое окно и открывать его в недиалоговом режиме, используя метод Show() (как в Блокноте).

5.2 Создание экрана-заставки

5.2.1 Добавить в приложение из п.5.1 изображение, которое должно отображаться перед запуском, пока загружаются ресурсы приложения (как в Visual Studio 2022).

5.2.2 Настроить свойства изображения:

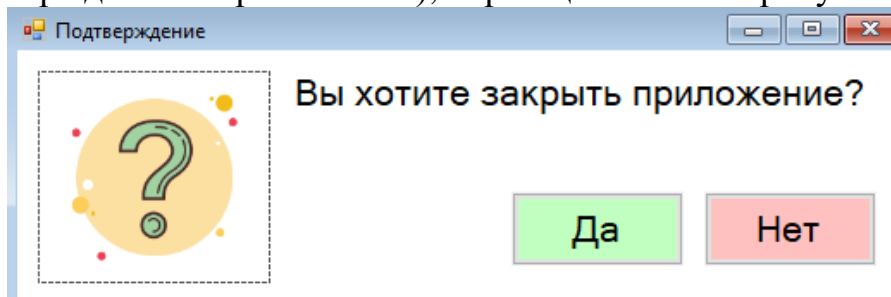
- Действия при сборке: SplashScreen

- Копировать в выходной каталог: более позднюю версию

5.2.3 Пересобрать проект.

5.3 Создание диалогового окна для подтверждения закрытия окна

5.3.1 Добавить в приложение из п.5.1 новое окно (аналог MessageBox для подтверждения закрытия окна), образец показан на рисунке:



5.3.2 Кнопка «Нет» должна срабатывать при нажатии Esc. Кнопка «Да» должна срабатывать при нажатии Enter. При нажатии «Да» изменять значение DialogResult окна на значение true.

5.3.3 Добавить в приложение обработчик события окна Closing. В обработчике события Closing открыть созданное окно подтверждения закрытия. Если пользователь не нажал «Да», то для отмены закрытия написать:

e.Cancel = true;

5.4 Создание диалоговых окон для разграничения прав доступа:

5.4.1 Создать WPF-приложение. В которое добавить:

- окно авторизации с полями ввода логина и пароля, кнопками «Войти» и «Отмена»,
- окно администратора с кнопкой «Назад» (должна закрывать окно),
- окно менеджера с кнопкой «Назад» (должна закрывать окно).

У всех окон указать:

- минимальный размер окна,
- положение при запуске (WindowStartupLocation) по центру экрана,
- русскоязычный заголовок.

5.4.2 На главное окно добавить кнопку «Авторизоваться».

При нажатии на кнопку «Авторизоваться»:

- скрыть главное окно (метод Hide()),
- открыть окно авторизации в диалоговом режиме,
- показать главное окно (метод Show()).

5.4.3 Если пользователь нажал «Войти» в окне авторизации, то:

- при значениях логин admin и пароль qwerty открывать окно администратора,
- при значениях логин manager и пароль 12345 открывать окно менеджера.

При корректных данных скрыть форму авторизации и открыть требуемое окно в диалоговом режиме. После закрытия — показать форму авторизации.

При некорректных данных после нажатия «Войти» сообщить об ошибке и оставить окно авторизации открытым.

5.5 Создание диалогового окна для выбора цвета

5.5.1 Создать WPF-приложение.

5.5.2 Добавить в приложение новое окно, на котором разместить:

- кнопку «ОК»,
- кнопку «Отмена»,
- 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.

Кнопка «Отмена» должна срабатывать при нажатии Esc.

Кнопка «ОК» должна срабатывать при нажатии Enter.

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)

В окне выбора цвета показывать текущее значение, используя заливку в элементе управления (например, в Rectangle):

объект.Background = new SolidColorBrush(цвет);

5.5.4 Реализовать открытие окна со слайдерами из главного окна.

Если пользователь нажал в окне выбора цвета «ОК», то изменять цвет фона главного окна.

6 Порядок выполнения работы

6.1 Выполнить все задания из п.5 в одном решении LabWork36. 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 Как сделать доступными данные пользовательского диалогового окна вызывающим его окнам?

8.2 Какие значения может принимать переменная DialogResult?

8.3 Как открыть окно в диалоговом режиме?

8.4 Как открыть окно в недиалоговом режиме?

8.5 В чем отличие между диалоговым и недиалоговым режимами работы?