

## Практическая работа №8

### Диалоговые окна

**Задание:** Элементом списка является информация о человеке, включающая ФИО, серию паспорта, номер паспорта (1-999999). Реализовать добавление и редактирование данных человека с использованием диалогового окна, а также удаление данных человека.

Для решения поставленной задачи на основной форме (Main\_F) потребуются компоненты:

- Spisok\_L класса Label для подписи компонента Spisok\_LB;
- Spisok\_LB класса ListBox для хранения и отображения списка людей;
- Add\_B класса Button для активизации добавления данных о человеке;
- Change\_B класса Button для активизации редактирования данных выбранного человека;
- Del\_B класса Button для активизации удаления данных выбранного человека.

Установим следующие значения свойств компонентов (таблица 5):

Таблица 5 – Значения свойств компонентов основной формы для примера программы, использующей диалоговые окна

Компонент.Свойство	Значение
Spisok_L.Text	Спи&сок людей
Add_B.Text	Доб&авить
Change_B.Text	И&зменить
Del_B.Text	Уда&лить

На форме диалогового окна (People\_F) потребуются компоненты:

- FIO\_L класса Label для подписи компонента FIO\_TB;
- FIO\_TB класса TextBox для ввода ФИО человека;
- Series\_L класса Label для подписи компонента Series\_TB;
- Series\_TB класса TextBox для ввода серии паспорта человека;
- Number\_L класса Label для подписи компонента Number\_TB;
- Number\_TB класса TextBox для ввода номера паспорта человека;
- Ok\_B класса Button для подтверждения добавления или редактирования;
- Cancel\_B класса Button для отмены добавления или редактирования.

Установим следующие значения свойств компонентов (таблица 6):

Таблица 6 – Значения свойств компонентов диалогового окна для примера программы, использующей диалоговые окна

Компонент.Свойство	Значение
People_F.FormBorderStyleStyle	FixedDialog
People_F.AcceptButton	Ok_B
People_F.CancelButton	Cancel_B
People_F.MinimizeBox	false
People_F.MaximizeBox	false
People_F.ShowInTaskbar	false
FIO_L.Text	ФИО &человека
Series_L.Text	Се&рия паспорта
Number_L.Text	Но&мер паспорта
Ok_B.Text	Ok
Ok_B.DialogResult	OK
Cancel_B.Text	Отм&ена
Cancel_B.DialogResult	Cancel

Опишем событие FormClosing формы People\_F:

```
private void People_F_FormClosing(object sender,
                                FormClosingEventArgs e)

{
    /* Проверку следует проводить только при нажатии
       кнопки "Ok" */
    if (DialogResult==DialogResult.OK)
        /* Будем использовать обработку исключительных ситуаций
           для проверки на правильность ввода и вывода информации
           об ошибках */
        try
        {
            /* Проверка ФИО */
            if (FIO_TB.Text.Trim() == "") // Если ничего не введено
            {
                /* Перевод фокуса ввода на компонент FIO_TB */
                FIO_TB.Focus();
                /* Вызов исключительной ситуации с текстом ошибки */
                throw new Exception("Вы не ввели ФИО человека");
            }
            /* Проверка серии паспорта */
            if (Series_TB.Text.Trim() == "")
            {
                Series_TB.Focus();
                throw new Exception("Вы не ввели серию паспорта");
            }
            /* Получение значения номера паспорта */
            int n = Convert.ToInt32(Number_TB.Text);
            /* Проверка значения номера паспорта на вхождение в
               заданный диапазон */
            if (n < 1 || n > 999999)
            {
                Number_TB.Focus();
                throw new Exception("Номер паспорта должен быть в
                                   диапазоне 1..999999");
            }
        }
}
```

```

,
/* Если введенное значение номера паспорта не целое число,
   то обработка данной ошибки */
catch (FormatException)
{
    e.Cancel = true;
    Number_TB.Focus();
    MessageBox.Show(
        "Номер паспорта должен быть целым числом", "Ошибка");
}
/* Обработка всех других ошибок */
catch (Exception E)
{
    e.Cancel = true;
    MessageBox.Show(E.Message, "Ошибка"); // E.Message -
                                           текст ошибки
}
}
}

```

Опишем событие Shown формы People\_F:

```

private void People_F_Shown(object sender, EventArgs e)
{
    /* При открытии формы переход на компонент FIO_TB */
    FIO_TB.Focus();
}

```

Опишем пользовательский класс TPeople в отдельном модуле, который добавим в программу через меню «Проект → Добавить новый элемент → Файл с текстом программы». Класс должен быть описан в том же пространстве имен, что и основная программа. Опишем класс следующим образом:

```

namespace ...
{
    public class TPeople
    {
        public string FIO, Series;
        public int Number;
        public override string ToString()
        {
            return FIO;
        }
    }
}

```

Опишем событие Click кнопки Add\_B:

```
private void Add_B_Click(object sender, EventArgs e)
{
    /* Перед работой с диалоговой формой ее нужно создать */
    People_F Dialog = new People_F();
    /* Изменение заголовка диалогового окна */
    Dialog.Text = "Добавление человека";
    /* Установка значений полей ввода диалогового окна в
       значение "по умолчанию" */
    Dialog.FIO_TB.Text = "";
    Dialog.Series_TB.Text = "";
    Dialog.Number_TB.Text = "";
    /* Вызов диалогового окна и проверка результата завершения
       его работы. Если окно закрыто с подтверждением ввода
       данных, то выполнение действий по созданию нового
       элемента */
    if (Dialog.ShowDialog() == DialogResult.OK)
    {
        /* Создание экземпляра класса */
        TPeople People = new TPeople();
        /* Заполнение свойств экземпляра класса */
        People.FIO = Dialog.FIO_TB.Text.Trim();
        People.Series = Dialog.Series_TB.Text.Trim();
        People.Number = Convert.ToInt32(Dialog.Number_TB.Text);

        /* Добавление нового элемента в список людей */
        Spisok_LB.Items.Add(People);
    }
}
```

Опишем событие Click кнопки Change\_B:

```
private void Change_B_Click(object sender, EventArgs e)
{
    /* Проверка, выбран ли элемент для редактирования */
    if (Spisok_LB.SelectedIndex != -1)
    {
        /* Перед работой с диалоговой формой ее нужно создать */
        People_F Dialog = new People_F();
        /* Установка заголовка диалогового окна */
        Dialog.Text = "Изменение данных человека";
        /* Получение редактируемого элемента */
        TPeople People = Spisok_LB.SelectedItem as TPeople;
        /* Установка значений полей ввода диалогового окна в
           текущие значения редактируемого элемента */
        Dialog.FIO_TB.Text = People.FIO;
        Dialog.Series_TB.Text = People.Series;
        Dialog.Number_TB.Text = People.Number.ToString();
    }
}
```

```

    /* Вызов диалогового окна и проверка результата завершения
    его работы. Если окно закрыто с подтверждением ввода
    данных, то выполнение действий по редактированию
    элемента */
    if (Dialog.ShowDialog() == DialogResult.OK)
    {
        /* Изменение свойств редактируемого элемента */
        People.FIO = Dialog.FIO_TB.Text.Trim();
        People.Series = Dialog.Series_TB.Text.Trim();
        People.Number = Convert.ToInt32(Dialog.Number_TB.Text);
        /* Изменение строки в списке за счет изменения объекта,
        связанного со строкой */
        Spisok_LB.Items[Spisok_LB.SelectedIndex] = People;
    }
}
else
    /* Если элемент не выбран, то вывод сообщения */
    MessageBox.Show("Не выбран элемент в списке",
        "Информация");
}

```

Опишем событие Click кнопки Del\_B:

```

private void Del_B_Click(object sender, EventArgs e)
{
    /* Проверка, выбран ли элемент для удаления */
    if (Spisok_LB.SelectedIndex != -1)
        /* Удаление строки из списка */
        Spisok_LB.Items.RemoveAt(Spisok_LB.SelectedIndex);

    else
        /* Если элемент не выбран, то вывод сообщения */
        MessageBox.Show("Не выбран элемент в списке",
            "Информация");
}

```

Внешний вид и пример работы программы показаны на рисунке 6.

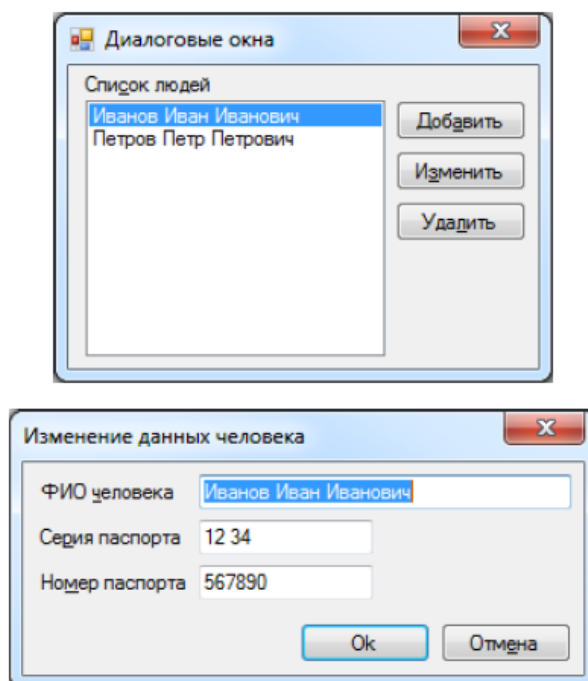


Рисунок 6 – Внешний вид и пример работы программы (основная форма и диалоговое окно), реализующей работу с диалоговыми окнами