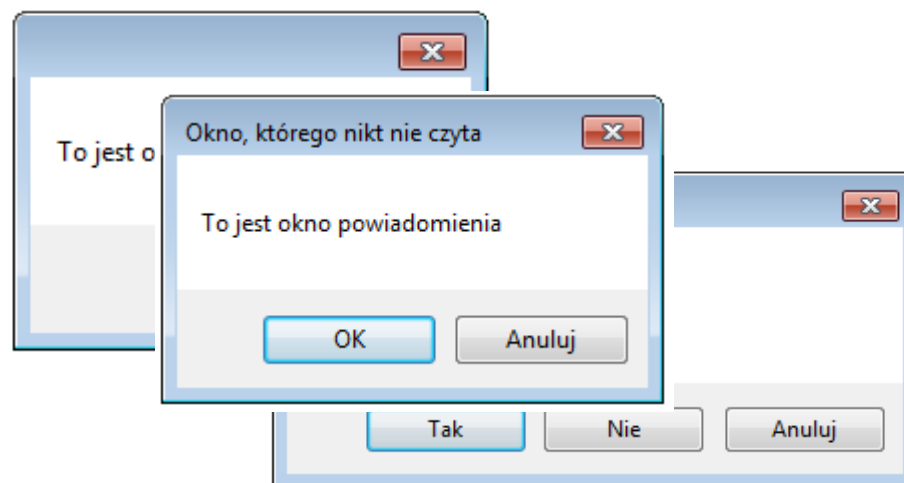


Wykład 7

Okna dialogowe, menu, pasek stanu itp.



Okna dialogowe

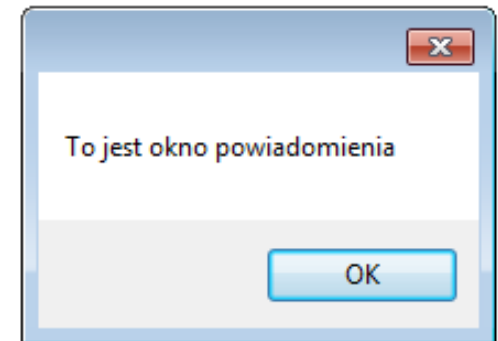


Okna dialogowe - komunikaty

Okno komunikatu to okno dialogowe, które może służyć do wyświetlania informacji tekstowych. Pozwala ono także użytkownikom na podejmowanie decyzji za pomocą przycisków.

```
MessageBox.Show("To jest okno powiadomienia");
```

W „standardzie” dostajemy tekst i przycisk „OK.”



Okno dialogowe możemy uzupełniać o różne elementy.

Okna dialogowe - komunikaty

Okno dialogowe możemy uzupełniać o różne elementy.

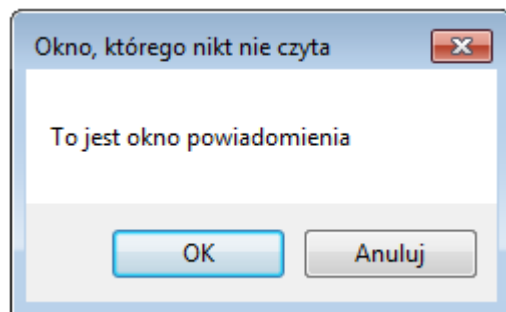
```
MessageBox.Show("To jest okno powiadomienia",
    this.Text="Okno, którego nikt nie czyta",
    MessageBoxButtons.);
```

Nagłówek

Rodzaje przycisków

- AbortRetryIgnore
- OK
- OKCancel
- RetryCancel
- YesNo
- YesNoCancel

MessageB
The messa



```
MessageBox.Show("To jest okno powiadomienia",
    this.Text="Okno, którego nikt nie czyta",
    MessageBoxButtons.OKCancel,
    MessageBoxDefaultButton.);
```

- Button1
- Button2
- Button3

MessageB
The first b

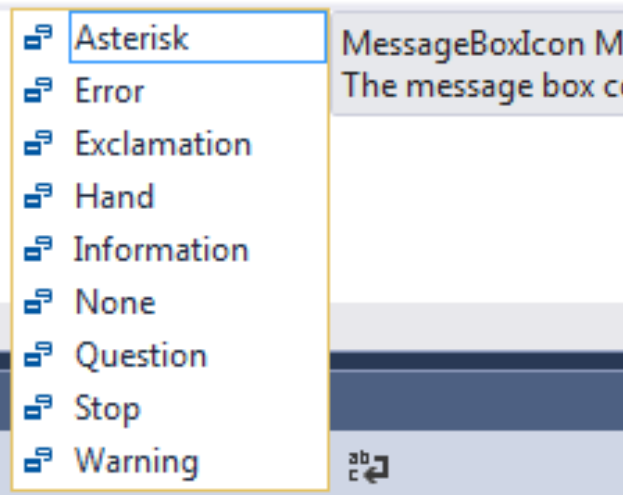
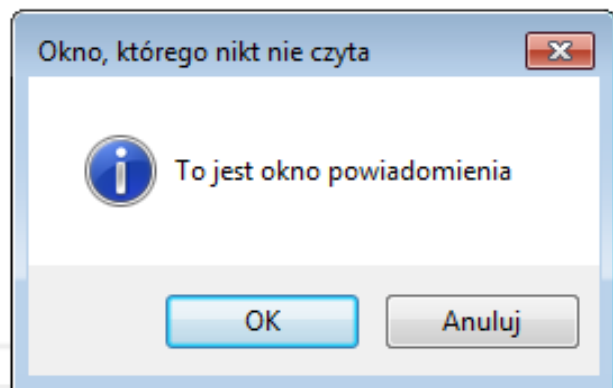
Defaultowy przycisk

Okna dialogowe - komunikaty

Okno dialogowe możemy uzupełniać o różne elementy.

```
MessageBox.Show("To jest okno powiadomienia",  
    this.Text="Okno, którego nikt nie czyta",  
    MessageBoxButtons.OKCancel,  
    MessageBoxDefaultButton.Button1,  
    MessageBoxIcon.  
);
```

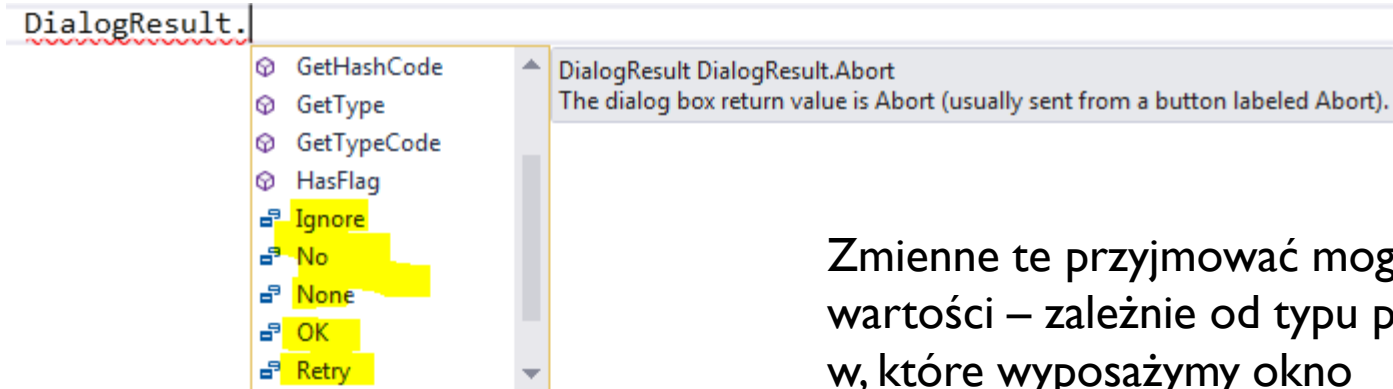
Wybór ikony



Okna dialogowe - komunikaty

Odpowiedź zwracaną przez okno (który przycisk naciśnięto) zapisać możemy w zmiennej typu `DialogResult`.

```
DialogResult odpowiedz = MessageBox.Show("To jest okno z pytaniem", "Pytanie",  
    MessageBoxButtons.YesNoCancel,  
    MessageBoxIcon.Question);
```

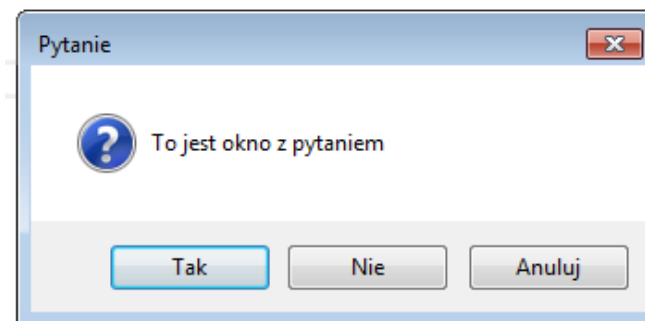


Zmienne te przyjmować mogą różne wartości – zależnie od typu przycisków w, które wyposażymy okno

Okna dialogowe - komunikaty

Okno o trzech możliwych odpowiedziach:

```
DialogResult odpowiedz = MessageBox.Show("To jest okno z pytaniem", "Pytanie",  
    MessageBoxButtons.YesNoCancel,  
    MessageBoxIcon.Question);  
  
switch (odpowiedz)  
{  
    case DialogResult.Yes: //akcja, jeżeli naciśnięto "TAK"  
        break;  
    case DialogResult.No: //akcja, jeżeli naciśnięto "NIE"  
        break;  
    case DialogResult.Cancel: //akcja, jeżeli naciśnięto "ANULUJ"  
        break;  
}
```

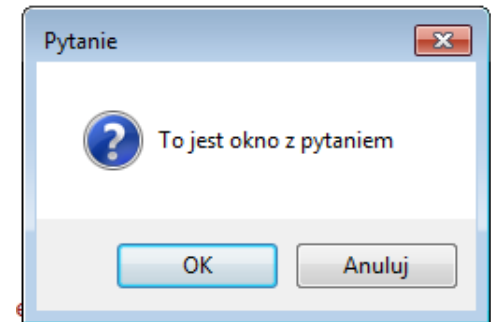


Okna dialogowe - komunikaty

Okno o 2 odpowiedziach:

```
DialogResult odpowiedz2 = MessageBox.Show("To jest okno z pytaniem", "Pytanie",  
    MessageBoxButtons.OKCancel,  
    MessageBoxIcon.Question);
```

```
if (odpowiedz2==DialogResult.OK)  
{  
    //Akcja, jeżeli naciśnięto "OK"  
}  
else  
{  
    //Akcja, jeżeli naciśnięto "ANULUJ"  
}
```



Zapisywanie odpowiedzi okna w pośredniczącej zmiennej nie jest konieczne.

```
if (MessageBox.Show("To jest okno z pytaniem", "Pytanie",  
    MessageBoxButtons.OKCancel, MessageBoxIcon.Question) == DialogResult.OK)  
{  
    //Akcja, jeżeli naciśnięto "OK"  
}  
else  
{  
    //Akcja, jeżeli naciśnięto "ANULUJ"  
}
```

Zapis
skrótowy:



Okna dialogowe - komunikaty

Przykład - potwierdzenie zamknięcia programu

```
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    DialogResult odpowiedz = MessageBox.Show("Jesteś pewien?", "Pytanie",
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Question);
    if (odpowiedz != DialogResult.Yes)
        e.Cancel = true;
```

„e” to zdarzenie przesłane w parametrze.

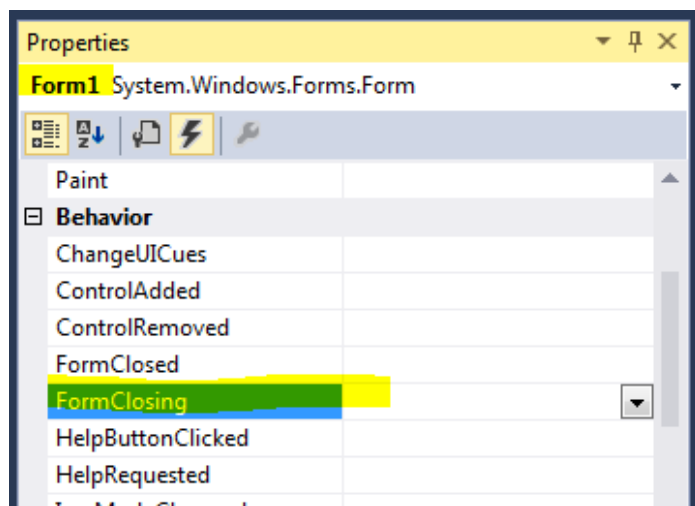
Ustawienie jego pola „Cancel” na „true” kasuje polecenie zamknięcia okna.

Okna dialogowe - komunikaty

Przykład - potwierdzenie zamknięcia programu

```
private void button1_Click(object sender, EventArgs e)
{
    Close();
}
```

Do zamknięcia okna (w przypadku okna głównego równoznaczne z zamknięciem programu) służy polecenie `Close()`;



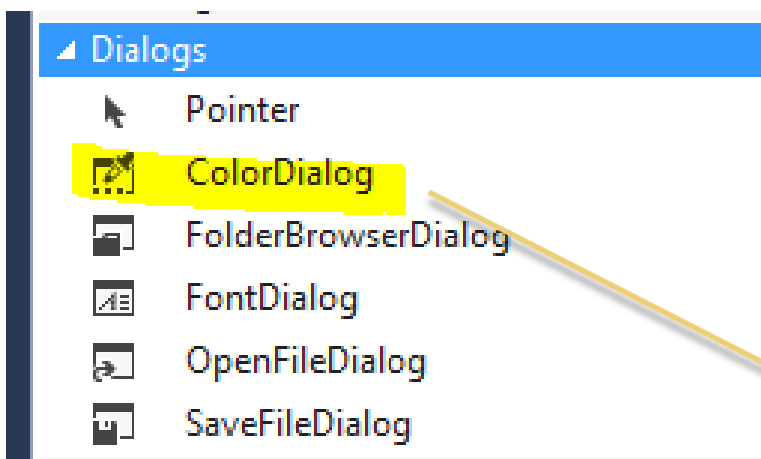
Aby przed zamknięciem okna wykonana została jakaś czynność (np. wyświetlenie okna dialogowego) oprogramować należy zdarzenie **FormClosing**



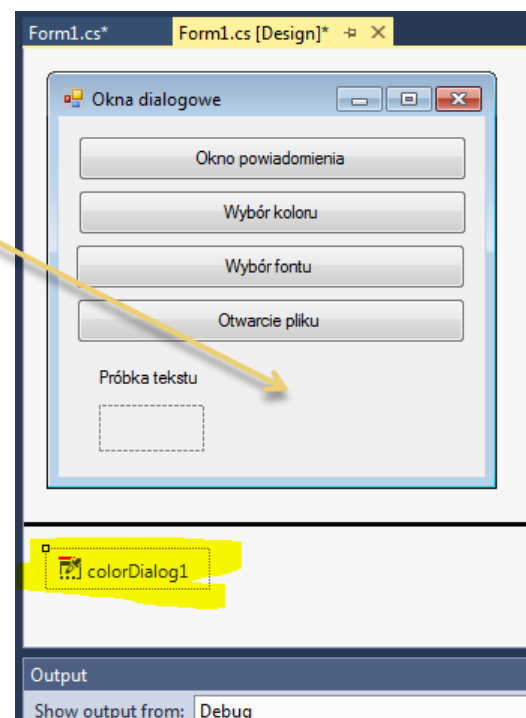
Okna dialogowe systemu

Okna dialogowe – **wybór koloru**

Krok I: dodanie do aplikacji obiektu klasy „ColorDialog”



Po przeciągnięciu na Form1 pojawi się on na liście obiektów niewidocznych (poniżej)

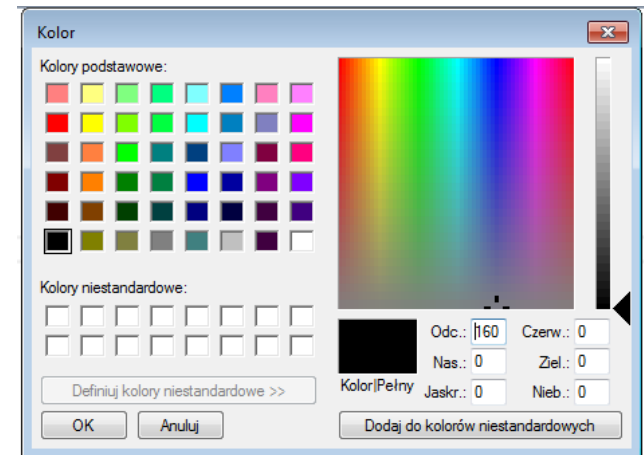


Okna dialogowe – wybór koloru

Krok 2: wywołanie metody „ShowDialog” uruchamia okno

```
colorDialog1.ShowDialog();
```

Po zamknięciu okna przyciskiem „OK” w polu obiektu „colorDialog1.Color” zapamiętana jest wybrana wartość koloru.

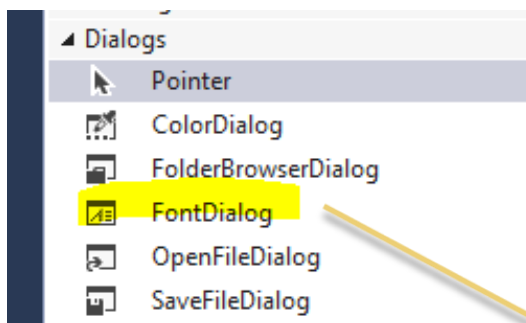


```
if (colorDialog1.ShowDialog() == DialogResult.OK)  
    panel1.BackColor = colorDialog1.Color;
```

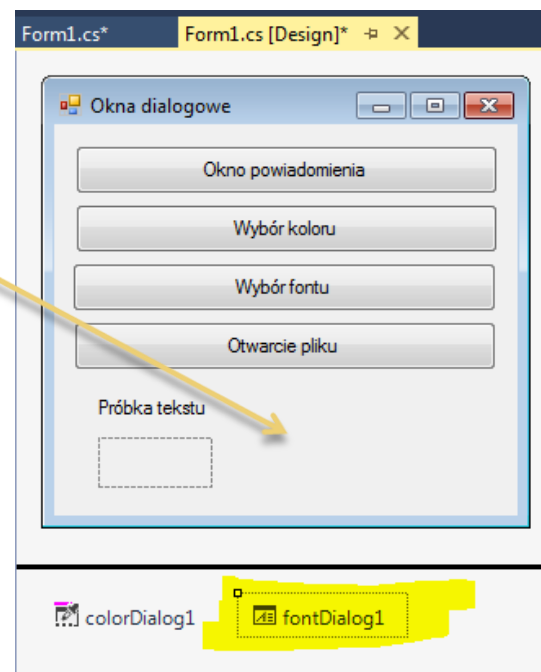
Okna dialogowe – **wybór fontu**

Krok I: dodanie do aplikacji obiektu klasy „FontDialog

”



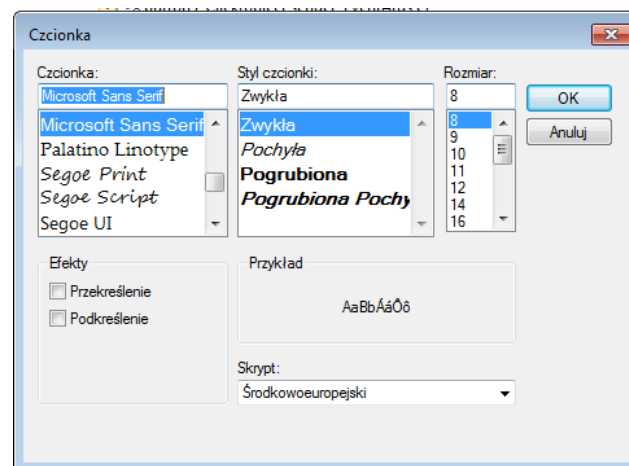
Po przeciągnięciu na Form1 pojawi się on na liście obiektów niewidocznych (poniżej)



Okna dialogowe – wybór fontu

Krok 2: wywołanie metody „ShowDialog” uruchamia okno

```
fontDialog1.ShowDialog();
```



Po zamknięciu okna przyciskiem „OK” w polu obiektu „fontDialog1.Font” zapamiętane są wybrane atrybuty fontu.

```
if (fontDialog1.ShowDialog() == DialogResult.OK)  
    label1.Font = fontDialog1.Font;
```

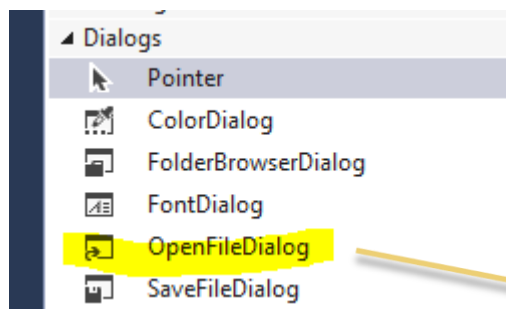
Okna dialogowe – Okno otwierania pliku

Na początek ważna informacja:

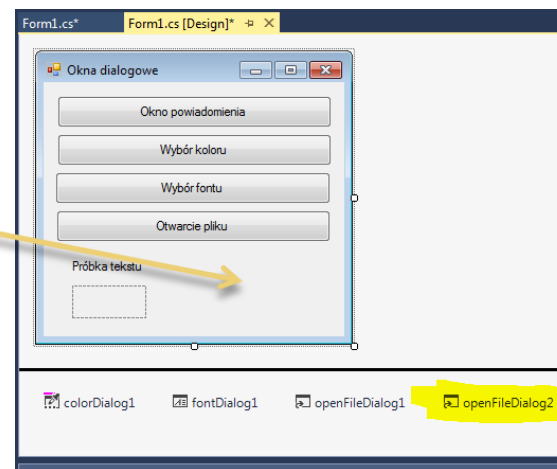
okno otwierania pliku żadnego pliku nie otworzy!

Okno (klasa OpenFileDialog) służy tylko do podania nazwy pliku i ścieżki.

Krok I: dodanie do aplikacji obiektu klasy „FontDialog”

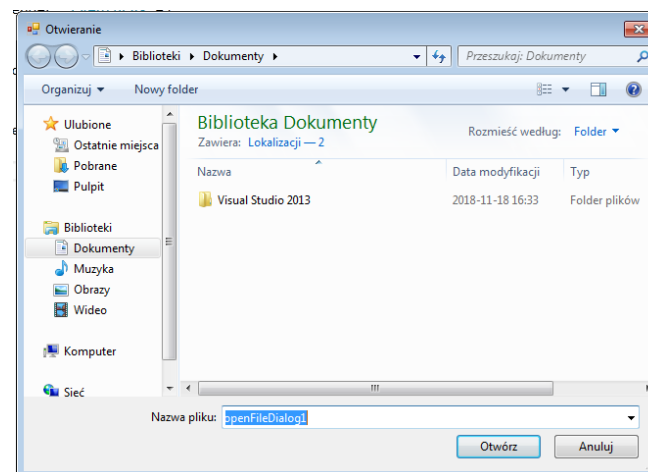


Po przeciągnięciu na Form1 pojawi się on na liście obiektów niewidocznych (poniżej)



Okna dialogowe – Okno otwierania pliku

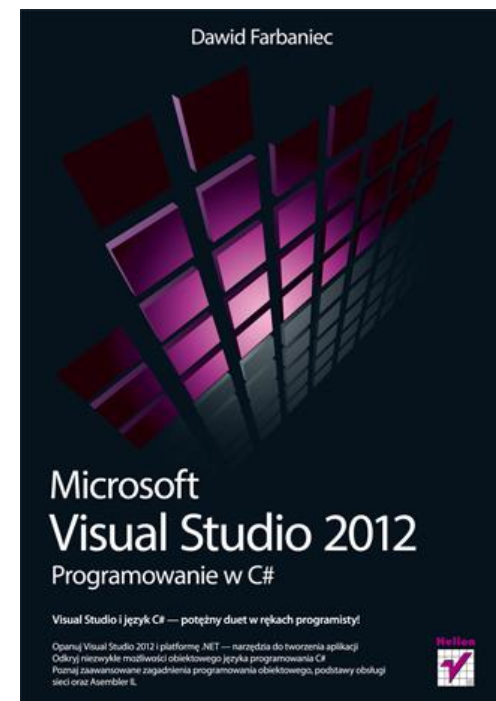
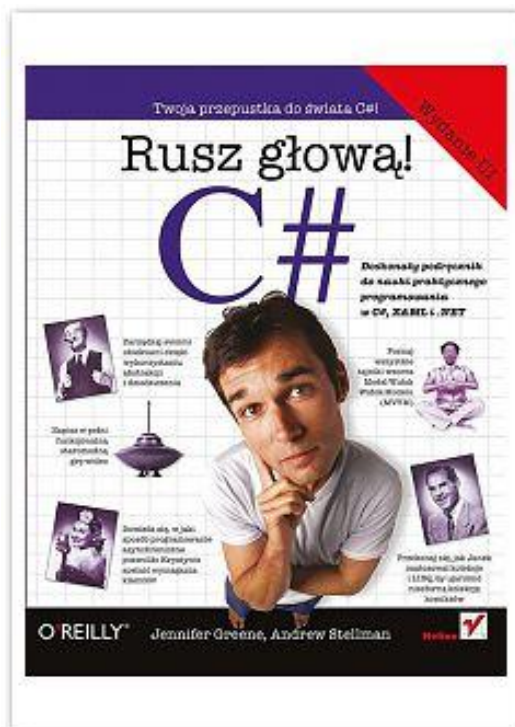
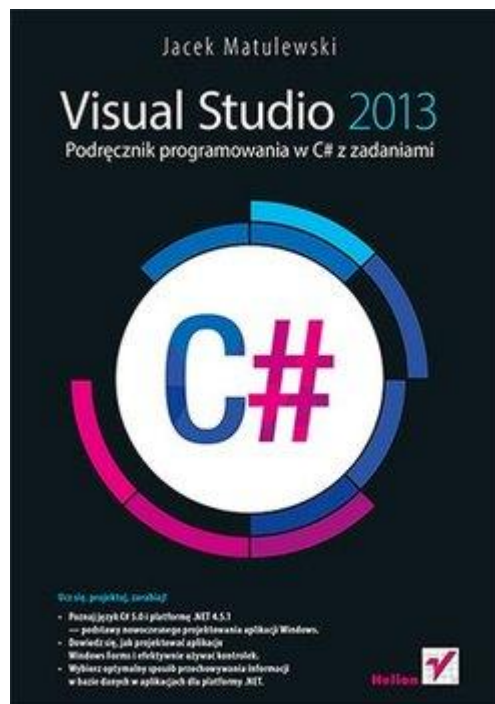
Krok 2: wywołanie metody „ShowDialog” uruchamia okno



Po zamknięciu okna przyciskiem „OK” w polu obiektu „openFileDialog.FileName” znajduje się ścieżka i nazwa wybranego pliku..

```
if (openFileDialog1.ShowDialog() == DialogResult.OK)
    label1.Text = openFileDialog1.FileName;
```

Literatura:



Użyte w tej prezentacji tabelki pochodzą z książki: Visual Studio 2013. Podręcznik programowania w C# z zadaniami
Autor: Matulewski Jęko, Helion