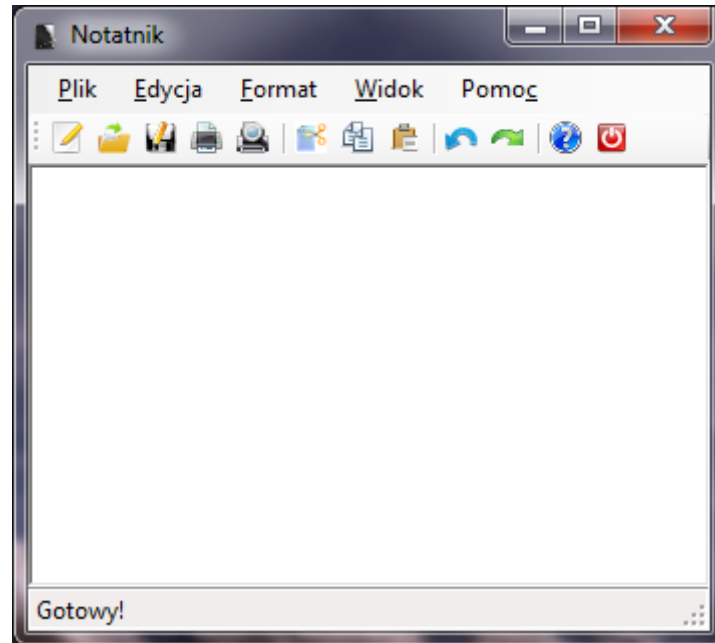


Notatnik w Windows Forms

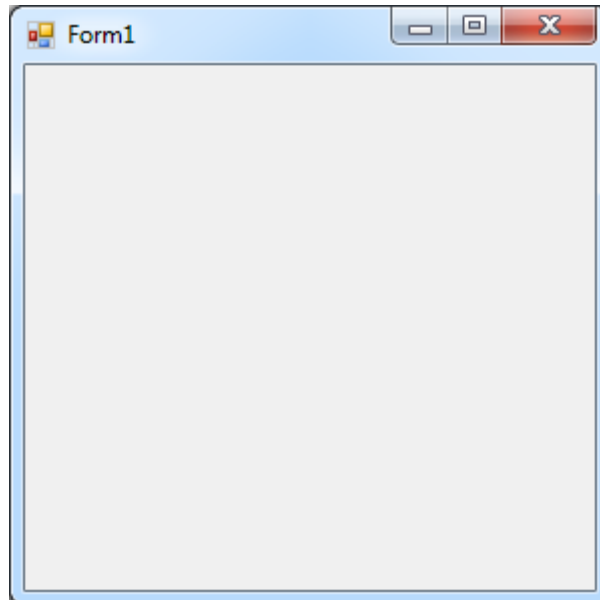
Dziś napiszemy prostą aplikację przypominającą **Notatnik** - standardowy składnik Windows. Nauczymy się jak dodać menu do programu, poznamy również metody postępowania z kontrolką **RichTextBox**. Dowiemy się także jak zapisać/odczytać tekst do i z pliku oraz wykorzystamy wbudowane okienka dialogowe.



Program jaki stworzymy w tym tutorialu

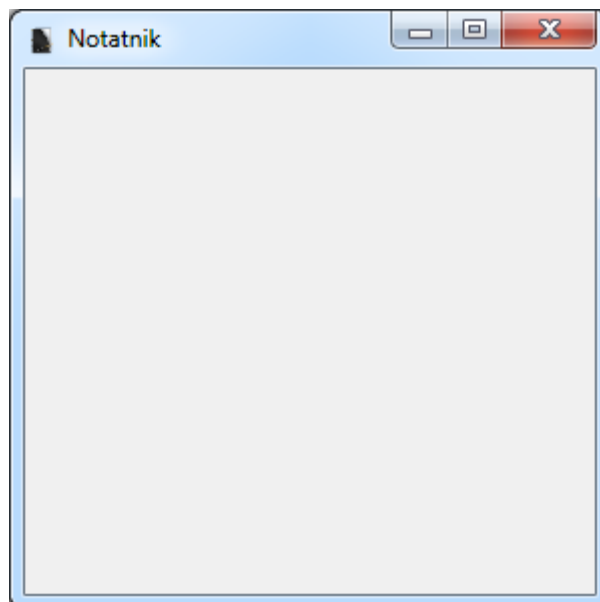
Okno główne

Wybieramy: File> New> Project... w nowym oknie wybieramy **Win32 Console Application** i w polu **Name** podajemy nazwę projektu (najlepiej taką, która dobrze opisuje co ten projekt zawiera). Klikamy OK, potem **Next>** i **Finish**.



Możemy, zmienić nazwę okna w **Properties** we właściwości **Text**, zamiast Form1 wpisujemy np. Notatnik. Dla okna programu możemy ustawić także ikonkę (we właściwości **Image** wskaż ikonkę).

Nasz program powinien wyglądać mniej więcej tak:



Pasek menu

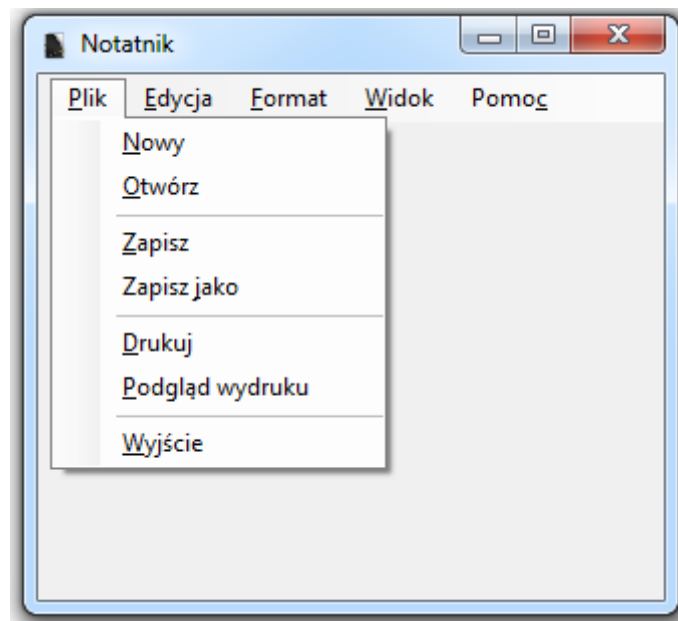
Z **Toolbox'a** (karta po prawej stronie, jeżeli jej nie ma: **View -> Toolbox** lub **Ctrl+Alt+X**) wybieramy komponent **MenuStrip** z grupy **Menus & Toolbars**.

Przeciągamy go na okno naszego programu. U góry okna pojawi się szare zaznaczenie. Klikamy w pole z napisem **Type Here** i wpisujemy na pierwszym miejscu Plik. Poniżej: Nowy, Otwórz, Zapisz, Zapisz Jako, Drukuj, Podgląd wydruku, Zamknij. Obok Plik tworzymy Edycja, Format, itd. zgodnie z poniższą tabelą

Plik	Edycja	Format	Widok	Pomoc
Nowy	Cofnij	Zawijanie wierszy	Pasek stanu	Wyświetl Pomoc
Otwórz	Ponów	Czcionka	Okno tekstowe	O programie...
Zapisz	Wytnij			
Zapisz jako	Kopiuj			
Drukuj	Wklej			
Podgląd wydruku	Zaznacz wszystko			
Wyjście	Wyczyść			
	Data/Godzina			

*aby uzyskać, separator w postaci poziomej linii wpisujemy "-" myślnik jako nazwę pozycji w menu

Powinniśmy uzyskać taki efekt:

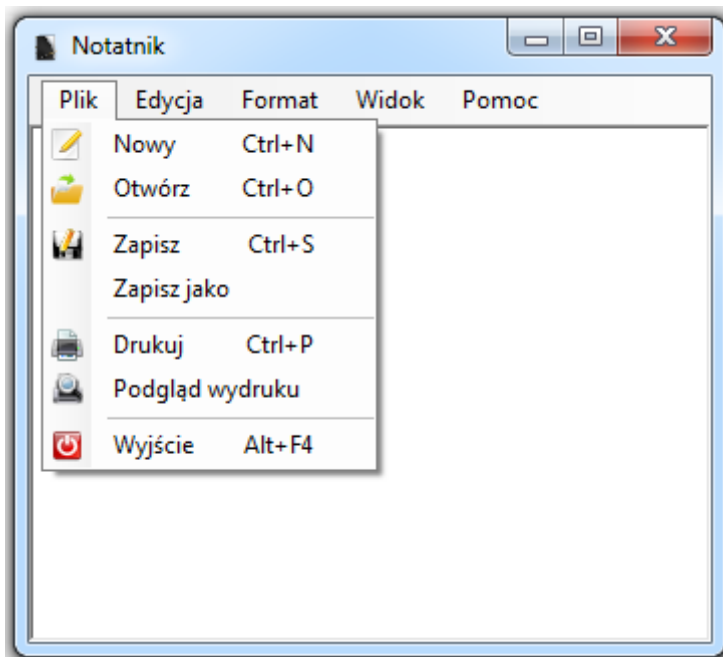


Skróty klawiszowe

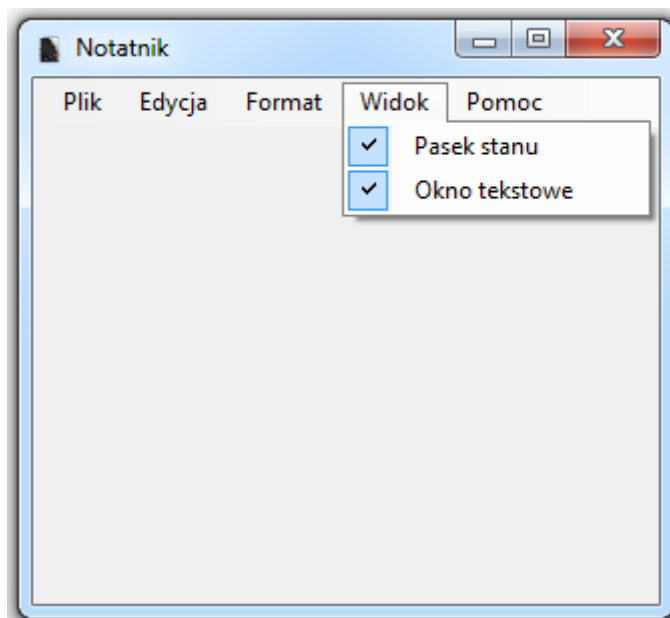
Dla niektórych opcji możemy podpiąć skróty klawiszowe - uruchamiane za pomocą lewego klawisza **Alt**. Aby to zrobić przed określoną literą w nazwie wstawiamy znak "&" lub/ oraz możemy dodać skróty dla określonych klawiszy poprzez zmianę w oknie **Properties** właściwości **ShortcutKeys**.

Ikony

Poprzez **Properties** we właściwości **Image** możemy, także dodać ikonkę dla danej pozycji w menu.



Dodatkowo dla **Format/Zawijanie wierszy**, **Widok/Pasek stanu** oraz **Widok/Okno tekstowe** w **Properties** ustawmy właściwość **Checked** w sekcji **Appearance** na wartość **True** oraz **CheckOnClick** w sekcji **Behavior** na wartość **True**.

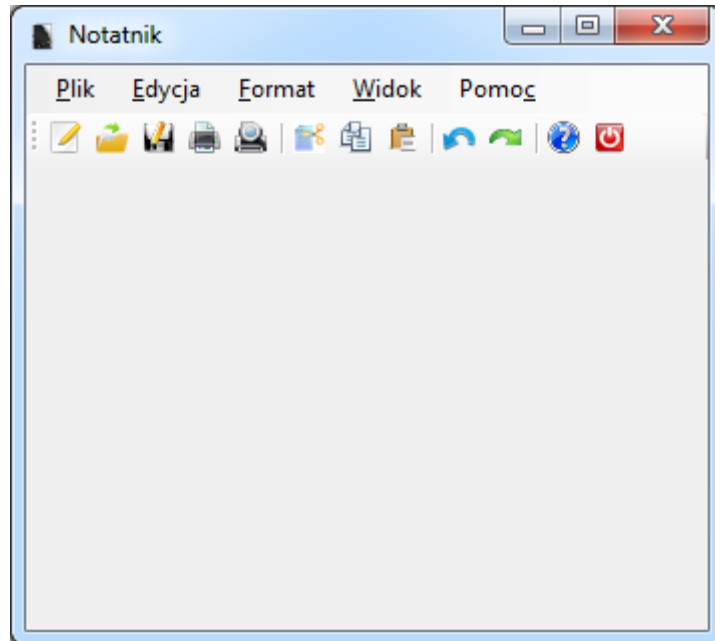


Pasek narzędziowy

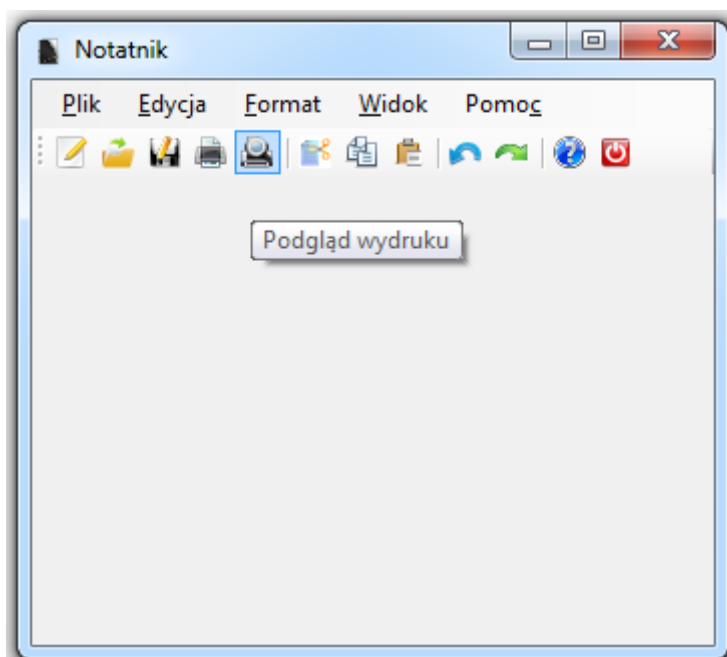
To najczęściej zbiór narzędzi, z których użytkownik często korzysta. Zwykle przedstawiany w formie ikonek.

Wybieramy w **Toolboxie** komponent **ToolStrip** i przeciągamy go na okno. Podobnie jak dodawaliśmy kolejne pozycje w menu tak i teraz dodajmy kilka przycisków.

Klikamy przycisk znajdujący się na dodanym pasku (nie rozwijamy listy). Pojawi się ikonka obrazu. W **Properties** dla właściwości **Image** wskazujemy ikonkę. Optymalny wybór to obraz 16×16 w formacie **ico**. W Internecie można znaleźć sporo ikonek, poszukaj, jedna z takich stron to IconFinder.com. Możemy także dodać separator poprzez wybranie go z listy rozwijanej.



Czasami ikonka może niewystarczająco wyjaśniać co się stanie po kliknięciu. Możemy więc dodać dymek z informacją, która pojawi się po najechnaniu myszką. Wystarczy ustawić (oczywiście dla każdego obrazka osobno) w **Properties** wartość dla **Text** np. Podgląd wydruku. Oto efekt:



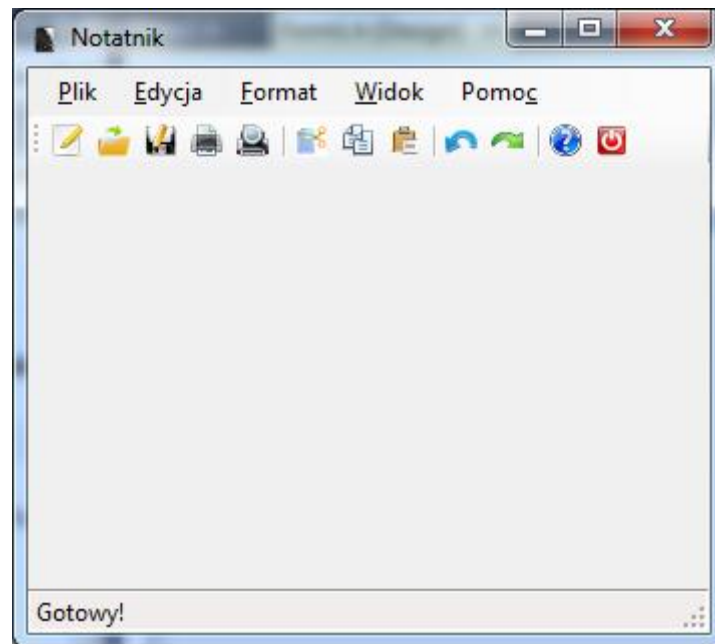
Pasek stanu

Pasek stanu pokazuje informacje o np. stanie operacji, pobieranym pliku itp. Jest wygodnym rozwiązaniem, gdyż nie ma sensu wyświetlać użytkownikowi okienek z informacjami typu: Operacja rozpoczęta! Operacja zakończona! Przygotowywanie ... Gotowy! itp.

Z **Toolbox'a** wybieramy komponent **StatusStrip** z grupy **Menus & Toolbars** i dodajemy go do programu.

Elementy wyświetlane dodajemy w sposób analogiczny jak w przypadku np. paska narzędzi. Tym razem możemy rozwinąć listę i wybrać etykietę **StatusLabel**. W **Properties** zmienimy właściwość **Text** na **Gotowy!**

Efekt:

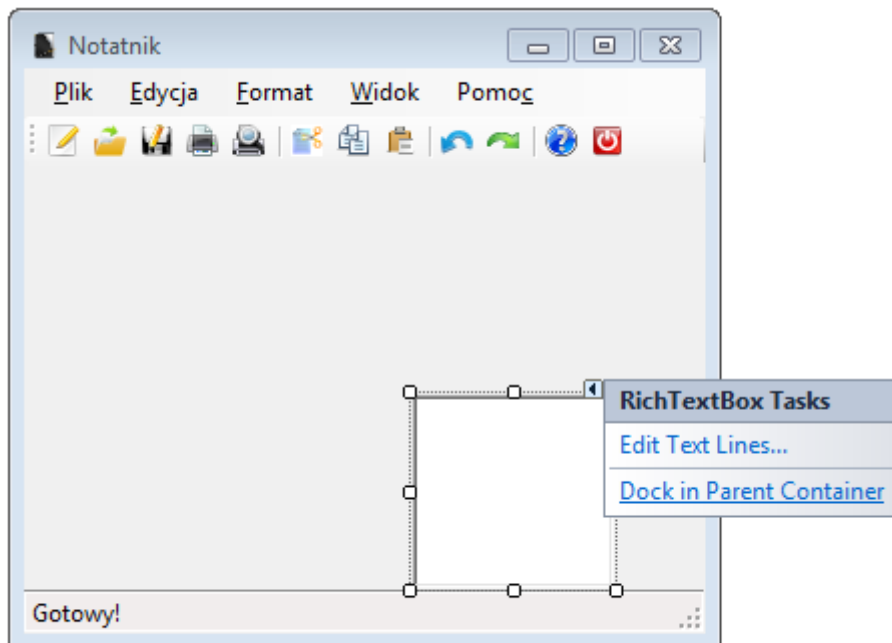


Dodajmy jeszcze trzy etykiety **StatusLabel**, a w **Properties** ustawmy dla nich właściwość **Text** na pusty ciąg znaków "", te etykiety będą nam potrzebne w przyszłości do wyświetlania ilości znaków oraz wierszy wpisywanych przez użytkownika.

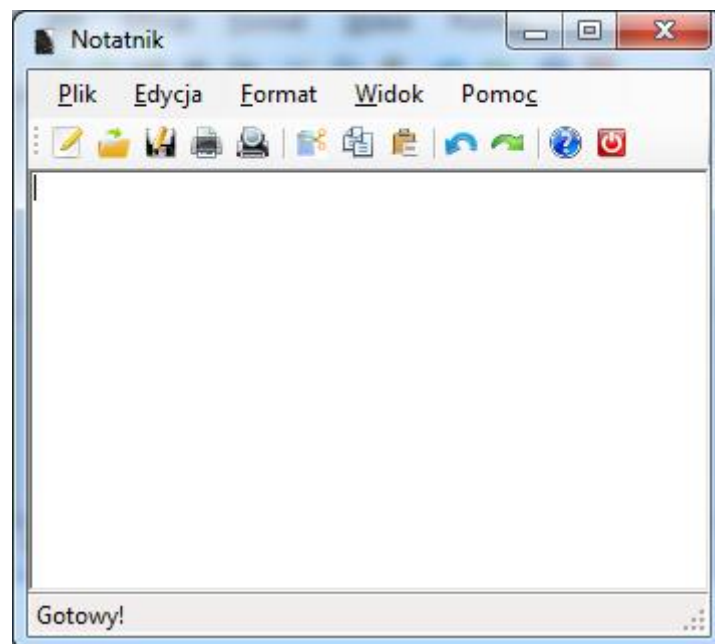
Pole tekstowe

W naszym programie brakuje jeszcze najważniejszego elementu, **Pola tekstowego**, aby je dodać **Toolbox'a** wybieramy komponent **RichTextBox** z grupy **Common Controls** i dodajemy go do programu.

Aby rozciągnąć je na cały obszar roboczy naszego programu musimy je zadokować. Dokonujemy tego klikając strzałkę i wybierając z wyświetlonego menu opcję **Dock in Parent Container**.



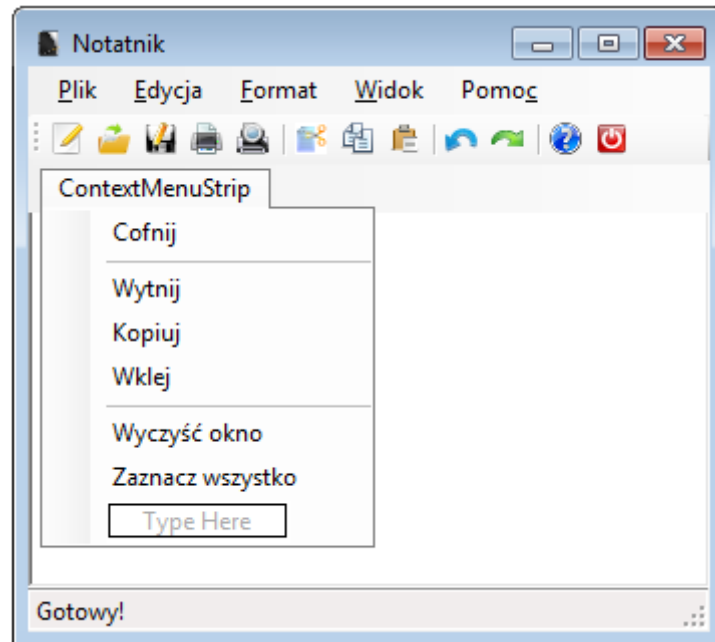
Nasz program powinien wyglądać tak:



Menu kontekstowe

Dla wszystkich, którzy nie wiedzą co to menu kontekstowe – jest to lista pojawiająca się po kliknięciu prawym przyciskiem myszy na dany obiekt.

Aby dodać do naszego programu tą funkcjonalność z **Toolbox'a** wybieramy komponent **ContextMenuStrip** z grupy **Menus & Toolbars** i dodajemy go do programu.



Tworzymy nasze menu kontekstowe, wpisując w kolejne pola nazwy poleceń. Aby uzyskać separator(poziomą kreskę) jako nazwę wpisujemy oczywiście myślnik "-".

Chcemy aby menu kontekstowe było wyświetlane po kliknięciu prawym przyciskiem myszy w obszarze pola tekstowego. Aby to zrobić wystarczy w oknie **Properties** dla **richTextBox1**(pola tekstowego) ustawić właściwość **ContextMenuStrip** na wartość **contextMenuStrip1**.

Dodawanie kodu

Nasze komponenty są „martwe” – klikanie w nie nic nie daje. Teraz zaczniemy ożywiać nasz program.

Plik/Nowy

W menu wybierz **Plik/Nowy** i kliknij na niego dwa razy – otworzy się nowa karta z kodem programu. Kursor automatycznie ustawia się w odpowiednim miejscu.

Wklej poniższy kod:

```
// Reset -> wyczyszczenie pola tekstowego
this->richTextBox1->Clear();

// Reset -> czcionka
this->richTextBox1->ResetFont();

// Reset -> kolor tekstu
this->richTextBox1->ResetForeColor();

// Reset -> zawijanie wierszy
this->zawijanieWierszyToolStripMenuItem->Checked = true;
this->richTextBox1->WordWrap = true;

// Reset -> widok/pole tekstowe
this->oknoTekstoweToolStripMenuItem->Checked = true;
this->richTextBox1->Show();

// Reset -> widok/pasek stanu
this->pasekStanuToolStripMenuItem->Checked = true;
this->statusStrip1->Show();
this->toolStripStatusLabel1->Text = "Gotowy!";
this->toolStripStatusLabel2->Text = "";
this->toolStripStatusLabel3->Text = "";
this->toolStripStatusLabel4->Text = "";
```

Plik/Otwórz

Z **Toolbox'a** wybieramy komponent **OpenFileDialog** z grupy **Dialogs** i dodajemy go do programu.

W **Properties** dla właściwości **Filter** ustawiamy jaki rodzaj plików zamierzamy otworzyć, wpisz - **Pliki Tekstowe (*.txt)|*.txt|All files (*.*)|*.***

W menu wybieramy **Plik/Otwórz** i klikamy na niego dwa razy – otworzy się nowa karta z kodem programu.

Wklej poniższy kod:

```
System::IO::Stream^ myStream;

if ( openFileDialog1->ShowDialog() == System::Windows::Forms::DialogResult::OK )
{
    if ( (myStream = openFileDialog1->OpenFile()) != nullptr )
    {
        this->richTextBox1->LoadFile( myStream, RichTextBoxStreamType::PlainText );
        myStream->Close();
    }
}
```

Plik/Zapisz

Z **Toolbox'a** wybieramy komponent **OpenFileDialog** z grupy **Dialogs** i dodajemy go do programu.

W **Properties** dla właściwości **Filter** ustawiamy jaki rodzaj plików zamierzamy otworzyć, wpisz - **Pliki Tekstowe (*.txt)|*.txt|All files (*.*)|*.***

W menu wybieramy **Plik/Otwórz** i klikamy na niego dwa razy – otworzy się nowa karta z kodem programu.

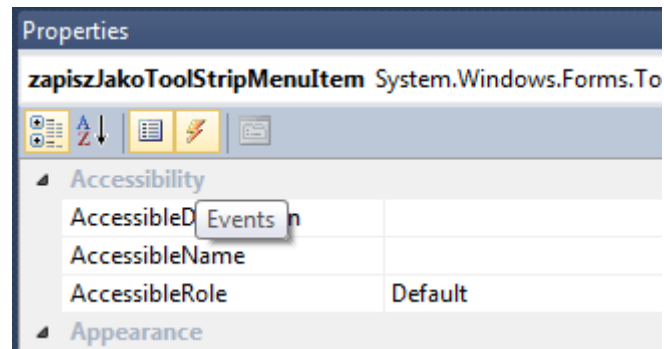
Wklej poniższy kod:

```
System::IO::Stream^ myStream;

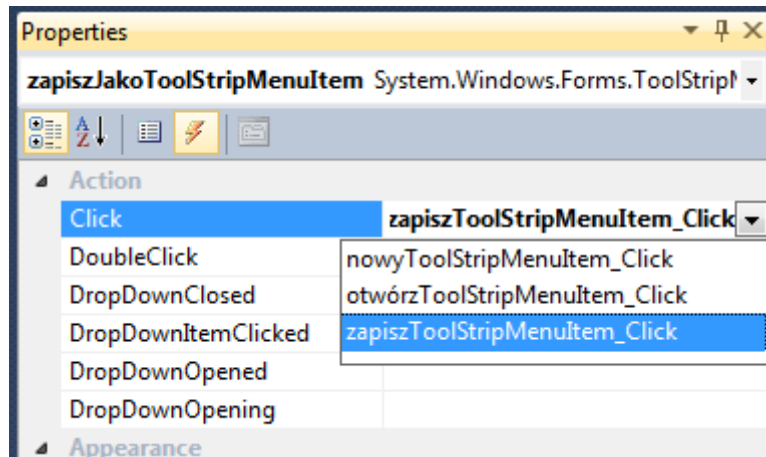
if ( saveFileDialog1->ShowDialog() == System::Windows::Forms::DialogResult::OK )
{
    if ( (myStream = saveFileDialog1->OpenFile()) != nullptr )
    {
        this->richTextBox1->SaveFile( myStream, RichTextBoxStreamType::PlainText );
        myStream->Close();
    }
}
```

Plik/Zapisz jako

Aby dodać akcję do tej pozycji w menu posłużymy się istniejącą już funkcją dla elementu **Plik/Zapisz**. Klikamy **raz** na element **Plik/Zapisz jako** w menu, a następnie w **Properties** przechodzimy na zakładkę **Events**.



dla właściwości **Click** wybieramy z listy rozwijanej funkcję Plik/Zapisz w moim przypadku ma ona nazwę `zapiszToolStripMenuItem_Click`.



Plik/Drukuj

Drukowanie jest trochę skomplikowane ze względu na konieczność ustawienia wszystkich parametrów – nie tylko tekstu ale także strony. Z tego też powodu nie można zrobić wszystkiego za pomocą jednej kontrolki.

Z **Toolbox'a** wybieramy następujące komponenty i dodajemy je do programu

- ✚ z grupy **Printing:**
 - **PrintDialog**
 - **PrintDocument**

- ✚ z grupy **Dialogs:**
 - **FontDialog**

W menu wybieramy **Plik/Drukuj** i klikamy na niego dwa razy – otworzy się nowa karta z kodem programu.

Wklejamy poniższy kod:

```
if (printDialog1->ShowDialog() == System::Windows::Forms::DialogResult::OK)
{
    stringToPrint = this->richTextBox1->Text;
    this->printDocument1->Print();
}
```

Następnie, klikamy dwa razy na dodany komponent **printDocument1** i wklejmy taki kod:

```
int charactersOnPage = 0;

int linesPerPage = 0;

StringFormat^ newStringFormat = gcnew StringFormat;

// Sets the value of charactersOnPage to the number
// of characters of stringToPrint that will fit within the bounds of the page.
e->Graphics->MeasureString( stringToPrint, this->fontDialog1->Font, e-
>MarginBounds.Size, newStringFormat, charactersOnPage, linesPerPage);

// Draws the string within the bounds of the page
e->Graphics->DrawString( stringToPrint, this->fontDialog1->Font, gcnew SolidBrush( this-
>fontDialog1->Color ), e->MarginBounds, newStringFormat);

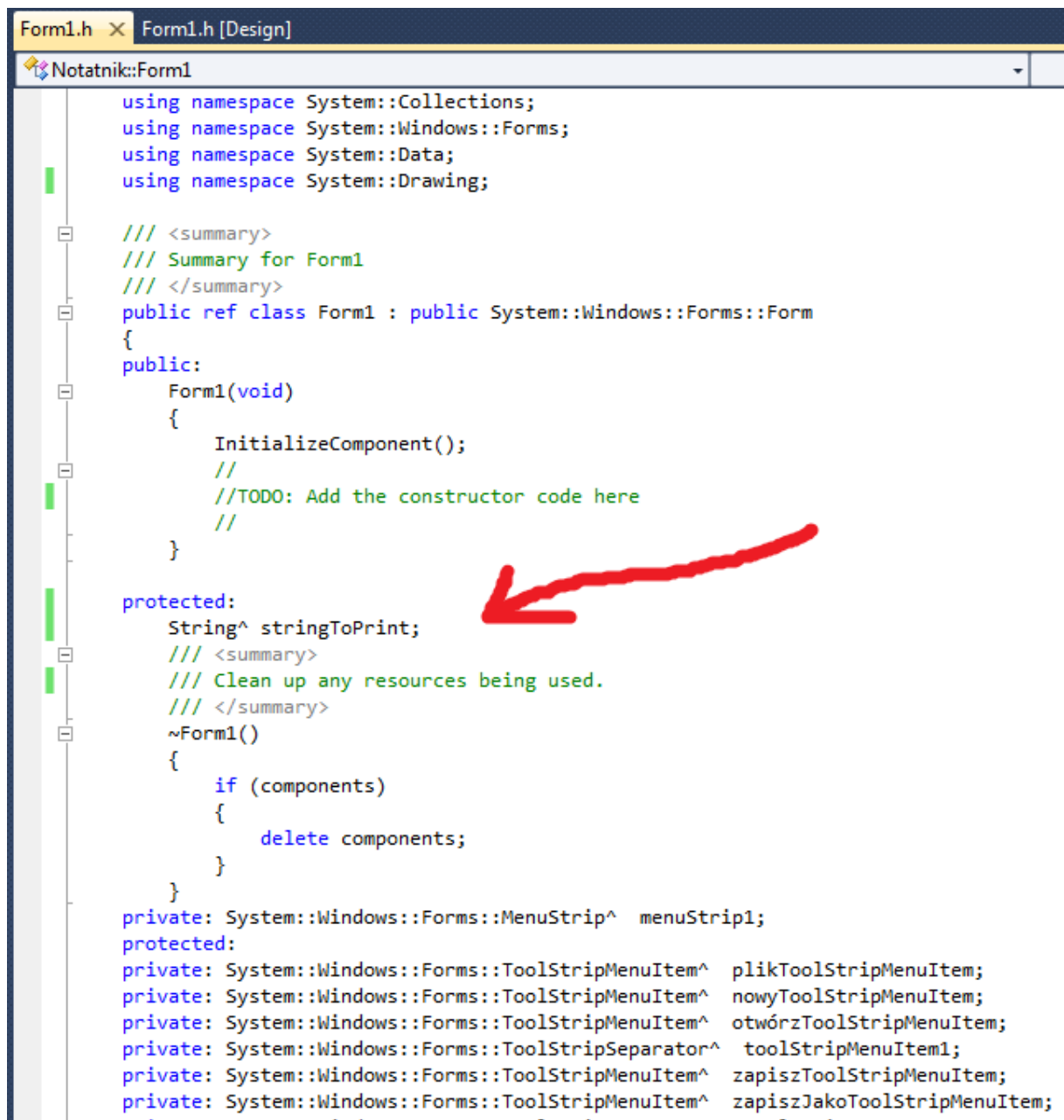
// Remove the portion of the string that has been printed.
stringToPrint = stringToPrint->Substring( charactersOnPage );

// Check to see if more pages are to be printed.
e->HasMorePages = (stringToPrint->Length > 0);
```

W kodzie musimy jeszcze tylko dodać zmienną tekstową **stringToPrint**, w której będziemy przechowywać tymczasowo tekst do wydruku.

Wklej poniższy kod na początku klasy **Form1** po etykiecie **protected:** tak jak na poniższym obrazku:

```
String^ stringToPrint;
```



```
Form1.h x Form1.h [Design]
Notatnik::Form1

using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;

/// <summary>
/// Summary for Form1
/// </summary>
public ref class Form1 : public System::Windows::Forms::Form
{
public:
    Form1(void)
    {
        InitializeComponent();
        //
        //TODO: Add the constructor code here
        //
    }

protected:
    String^ stringToPrint;
    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    ~Form1()
    {
        if (components)
        {
            delete components;
        }
    }

private: System::Windows::Forms::MenuStrip^ menuStrip1;
protected:
private: System::Windows::Forms::ToolStripMenuItem^ plikToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^ nowyToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^ otwórzToolStripMenuItem;
private: System::Windows::Forms::ToolStripSeparator^ toolStripMenuItem1;
private: System::Windows::Forms::ToolStripMenuItem^ zapiszToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^ zapiszJakoToolStripMenuItem;
```

Plik/Podgląd wydruku

Z **Toolbox'a** wybieramy komponent **PrintPreviewDialog** z grupy **Printing** i dodajemy go do programu.

W menu wybieramy **Plik/Podgląd wydruku** i klikamy na niego dwa razy – otworzy się nowa karta z kodem programu.

Wklejamy poniższy kod:

```
stringToPrint = this->richTextBox1->Text;  
printPreviewDialog1->Document = this->printDocument1;  
printPreviewDialog1->FormBorderStyle = System::Windows::Forms::FormBorderStyle::Fixed3D;  
printPreviewDialog1->ShowDialog();
```

Plik/Wyjście

W menu wybieramy **Plik/Wyjście** i klikamy na niego dwa razy – otworzy się nowa karta z kodem programu.

Wklejamy poniższy kod:

```
Application::Exit();
```

Edycja/Cofnij

W menu wybieramy **Edycja/Cofnij** i klikamy na niego dwa razy – otworzy się nowa karta z kodem programu.

Wklejamy poniższy kod:

```
if ( this->richTextBox1->CanUndo == true )  
    this->richTextBox1->Undo();
```

Edycja/Ponów

W menu wybieramy **Edycja/Ponów** i klikamy na niego dwa razy – otworzy się nowa karta z kodem programu.

Wklejamy poniższy kod:

```
if ( this->richTextBox1->CanRedo == true )  
    richTextBox1->Redo();
```

Edycja/Wytnij

W menu wybieramy **Edycja/Wytnij** i klikamy na niego dwa razy – otworzy się nowa karta z kodem programu.

Wklejamy poniższy kod:

```
this->richTextBox1->Cut();
```

Edycja/Kopiuj

W menu wybieramy **Edycja/Kopiuj** i klikamy na niego dwa razy – otworzy się nowa karta z kodem programu.

Wklejamy poniższy kod:

```
this->richTextBox1->Copy();
```

Edycja/Wklej

W menu wybieramy **Edycja/Wklej** i klikamy na niego dwa razy – otworzy się nowa karta z kodem programu.

Wklejamy poniższy kod:

```
this->richTextBox1->Paste();
```

Edycja/Zaznacz wszystko

W menu wybieramy **Edycja/Zaznacz wszystko** i klikamy na niego dwa razy – otworzy się nowa karta z kodem programu.

Wklejamy poniższy kod:

```
this->richTextBox1->SelectAll();
```

Edycja/Wyczyść

W menu wybieramy **Edycja/Wyczyść** i klikamy na niego dwa razy – otworzy się nowa karta z kodem programu.

Wklejamy poniższy kod:

```
this->richTextBox1->Clear();
```

Edycja/Data/Godzina

W menu wybieramy **Edycja/Data/Godzina** i klikamy na niego dwa razy – otworzy się nowa karta z kodem programu.

Wklejamy poniższy kod:

```
DateTime dateTime = DateTime::Now;  
  
this->richTextBox1->Text += dateTime.ToString();
```


Format/Zawijanie wierszy

W menu wybieramy **Format/Zawijanie wierszy** i klikamy na niego dwa razy – otworzy się nowa karta z kodem programu.

Wklejamy poniższy kod:

```
if( this->zawijanieWierszyToolStripMenuItem->Checked == true )
    this->richTextBox1->WordWrap = true;
else
    this->richTextBox1->WordWrap = false;
```

Format/Czcionka

Dla dodanego wcześniej elementu **fontDialog1** w **Properties** ustawiamy właściwość **ShowColor** na wartość **True**.

W menu wybieramy **Format/Czcionka** i klikamy na niego dwa razy – otworzy się nowa karta z kodem programu.

Wklejamy poniższy kod:

```
if(fontDialog1->ShowDialog() != System::Windows::Forms::DialogResult::Cancel )
{
    this->richTextBox1->Font = this->fontDialog1->Font ;
    this->richTextBox1->ForeColor = this->fontDialog1->Color;
}
```

Widok/Pasek stanu

W menu wybieramy **Widok/Pasek stanu** i klikamy na niego dwa razy – otworzy się nowa karta z kodem programu.

Wklejamy poniższy kod:

```
if( this->pasekStanuToolStripMenuItem->Checked == true )
    this->statusStrip1->Show();
else
    this->statusStrip1->Hide();
```

Widok/Okno tekstowe

W menu wybieramy **Widok/Okno tekstowe** i klikamy na niego dwa razy – otworzy się nowa karta z kodem programu.

Wklejamy poniższy kod:

```
if( this->oknoTekstoweToolStripMenuItem->Checked == true )
    this->richTextBox1->Show();
else
    this->richTextBox1->Hide();
```

Pomoc/Wyświetl pomoc

W menu wybieramy **Pomoc/Wyświetl pomoc** i klikamy na niego dwa razy – otworzy się nowa karta z kodem programu.

Wklejamy poniższy kod:

```
System::Diagnostics::Process::Start( "C://Windows//winhlp32.exe" );
```

Pomoc/O programie...

W menu wybieramy **Pomoc/O programie...** i klikamy na niego dwa razy – otworzy się nowa karta z kodem programu.

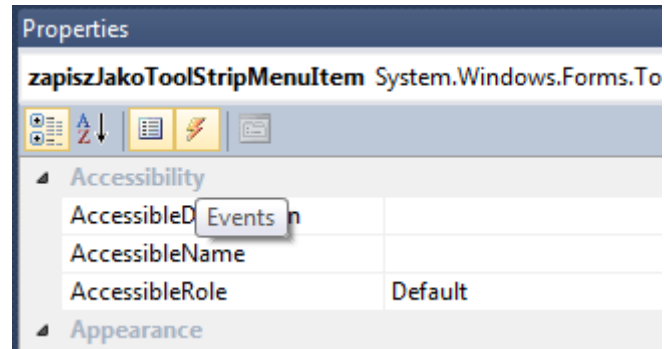
Wklejamy poniższy kod:

```
MessageBox::Show( "Program stworzony dzięki \nMicrosoft Visual C++ 2010 Express\n"
                  "Copyright © 2013 Michał Woźniak. All rights reserved. :)\n"
                  "Tutorial - Notatnik w Windows Forms",
                  "O programie...",
                  MessageBoxButtons::OK,
```

Pasek stanu

Mamy pasek stanu, ale nie jest on aktualizowany. Aby podczas pisania na pasku narzędzi były wyświetlane informacje o bieżącej ilości znaków oraz linii musimy obsłużyć zdarzenie **TextChanged**, które jest wywoływane podczas zmiany tekstu w polu tekstowym.

Otwieramy **Properties** dla **richTextBox1**(pola tekstowego), przechodzimy na zakładkę **Events**



odnajdujemy **TextChanged** i klikamy na nie dwa razy - otworzy się nowa karta z kodem programu.

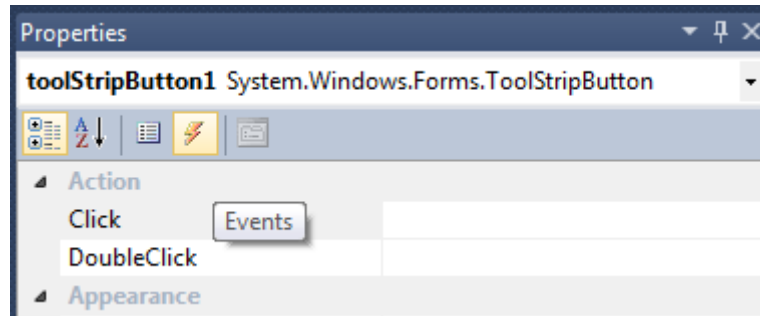
Wklejamy poniższy kod:

```
// *** Aktualizacja paska stanu ***  
  
// ilość znaków  
this->toolStripStatusLabel1->Text = "Znaków: ";  
this->toolStripStatusLabel2->Text = System::Convert::ToString( this->richTextBox1->TextLength );  
  
// ilość wierszy  
this->toolStripStatusLabel3->Text = "Wierszy: ";  
this->toolStripStatusLabel4->Text = System::Convert::ToString( (this->richTextBox1->Lines)->Length );
```

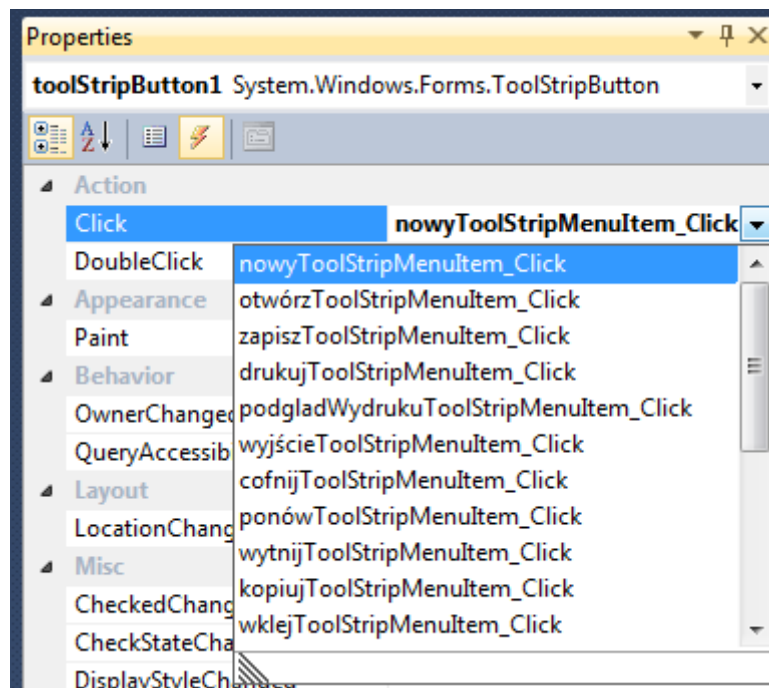
Pasek narzędziowy

Aby dodać akcję dla paska narzędziowego posłużymy się istniejącymi już funkcjami. Utworzyliśmy je dodając kod dla poszczególnych elementów menu.

Klikamy **raz** na pierwszej ikonkę na pasku narzędziowym, a następnie w **Properties** przechodzimy na zakładkę **Events**.



dla właściwości **Click** wybieramy z listy rozwijanej funkcję **odpowiednią** dla każdej z ikonek na pasku narzędziowym np. dla **Nowy** wybieramy **nowyToolStripMenuItem_Click**.

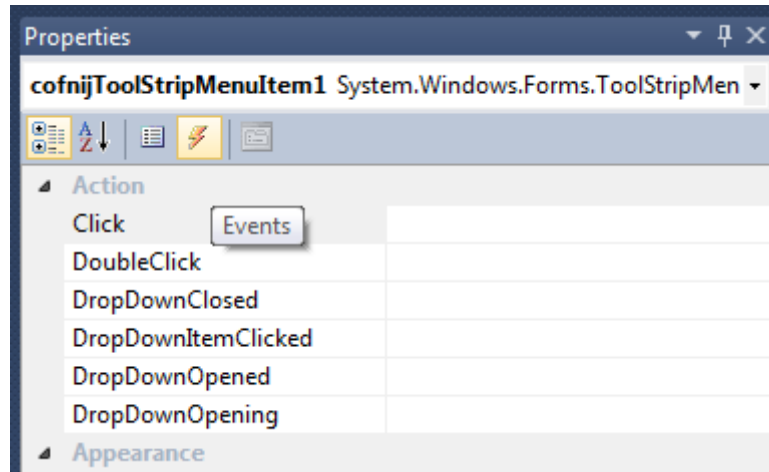


analogicznie ustawiamy funkcje dla reszty ikonki na pasku narzędziowym.

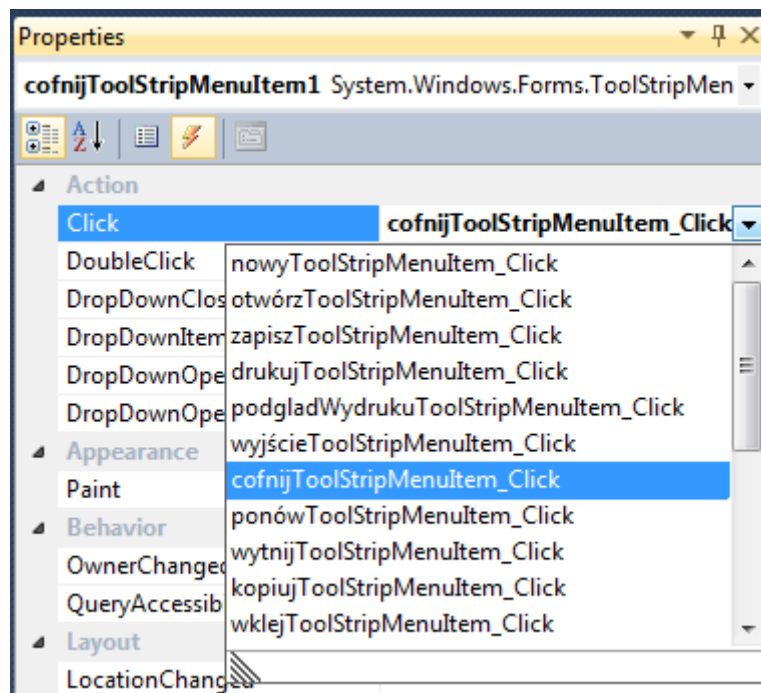
Menu kontekstowe

Aby dodać akcję dla menu kontekstowego tak jak w przypadku paska narzędziowego posłużymy się istniejącymi już funkcjami.

Otwieramy **Properties** dla obiektu **contextMenuStrip1**, klikamy raz na pierwszy element z menu kontekstowego **Cofnij** następnie przechodzimy na zakładkę **Events**.

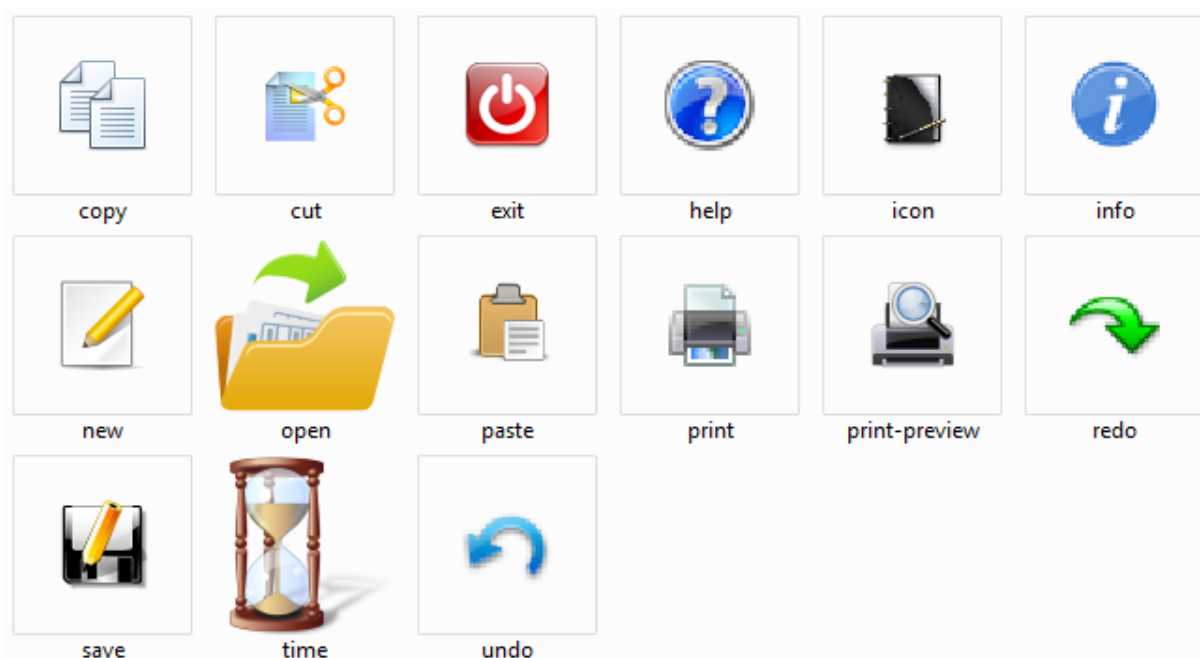


dla właściwości **Click** wybieramy z listy rozwijanej funkcję odpowiednią dla każdej z ikonek na pasku narzędziowym np. dla **Cofnij** wybieramy **cofnijToolStripMenuItem_Click**.



analogicznie ustawiamy funkcje dla reszty pozycji w menu kontekstowym.

Wykaz użytych ikon



Poniżej linki do ikon użytych w programie:

Ikona	Link
Copy	https://www.iconfinder.com/icons/173951/copy_icon#size=48
Cut	https://www.iconfinder.com/icons/17352/cut_icon#size=80
Exit	https://www.iconfinder.com/icons/3226/exit_power_quit_icon#size=128
Help	https://www.iconfinder.com/icons/34225/blue_circle_help_icon#size=128
Icon	https://www.iconfinder.com/icons/78822/notepad_icon#size=128
Info	https://www.iconfinder.com/icons/12825/info_icon#size=32
New	https://www.iconfinder.com/icons/9397/document_file_new_paper_pen_pencil_reply_write_icon#size=48
Open	https://www.iconfinder.com/icons/85334/file_open_icon#size=128
Paste	https://www.iconfinder.com/icons/20651/paste_icon#size=32
Print	https://www.iconfinder.com/icons/12308/print_printer_icon#size=24
Preview	https://www.iconfinder.com/icons/23217/document_preview_print_icon#size=128
Redo	https://www.iconfinder.com/icons/4089/redo_icon#size=32
Save	https://www.iconfinder.com/icons/7997/disk_edit_file_save_icon#size=64
Time	https://www.iconfinder.com/icons/34227/clock_history_hourglass_pending_time_waiting_icon#size=128
Undo	https://www.iconfinder.com/icons/44171/arrow_undo_icon#size=32

*ikony pochodzą z serwisu www.iconfinder.com