Wykład 5

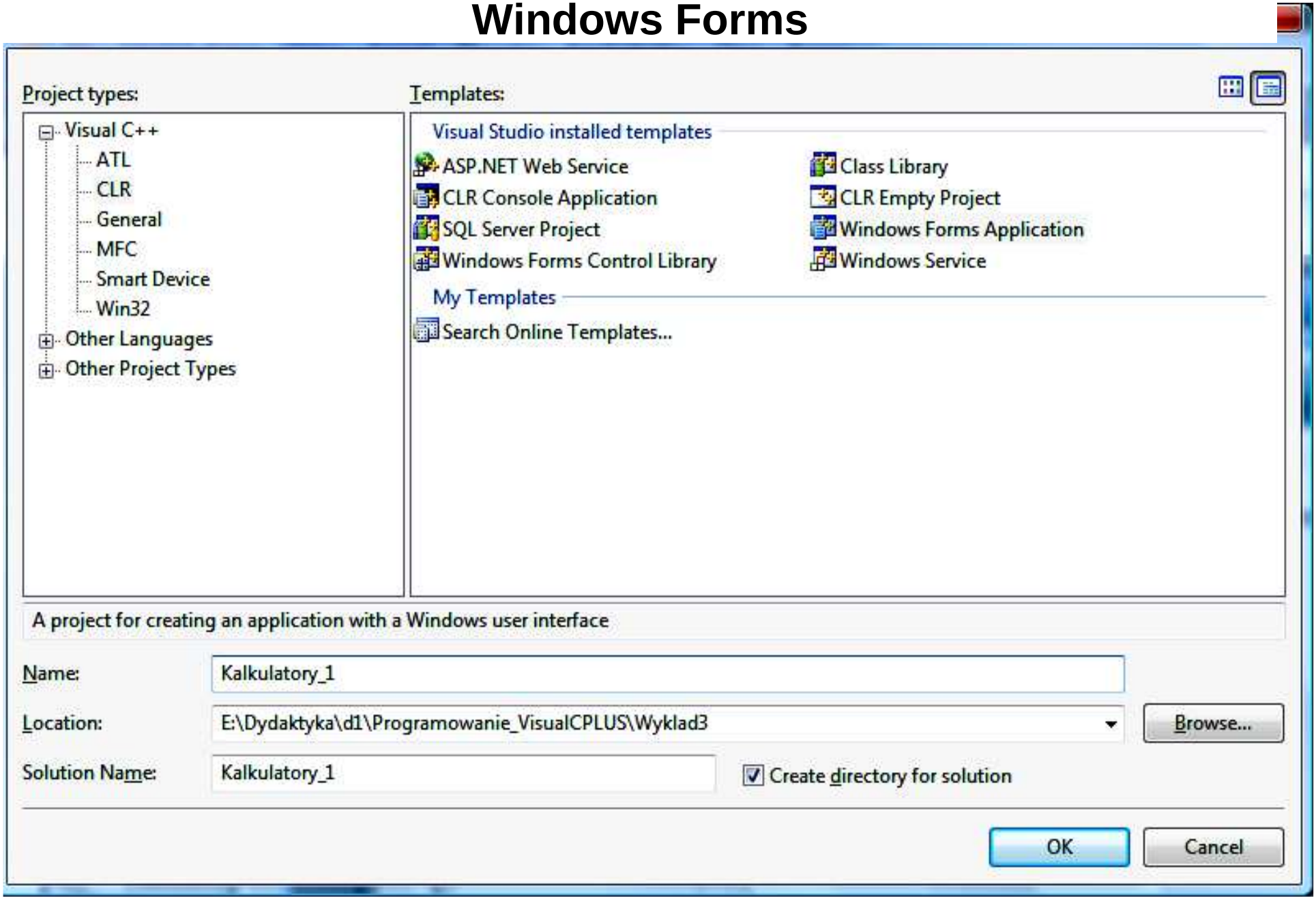
Okna MDI i SDI, dziedziczenie

Autor: Zofia Kruczkiewicz

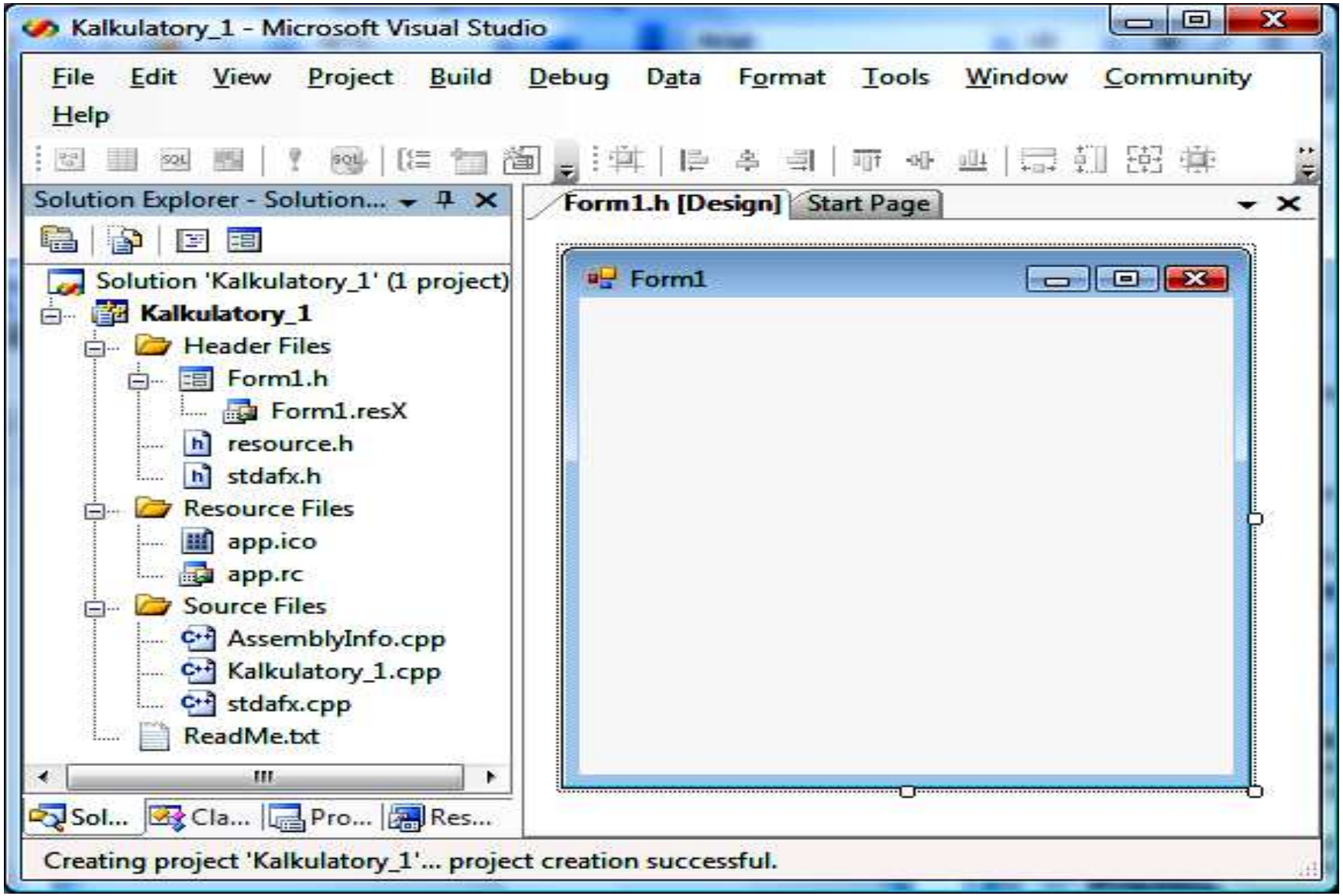
Zagadnienia

1. **Aplikacja wielookienkowa. Zakładanie projektu typu CLR Windows Forms**
   - **1.1. Aplikacja typu MDI**
   - **1.2. Aplikacja typu SDI**
2. **Dziedziczenie**
   - **2.1. Przykład metod abstrakcyjnych i klasy abstrakcyjnej typu ref - projekt konsolowy Figury**
   - **2.2. Przykład projektu konsolowego prezentującego rozwiązanie programu Figury z wykorzystaniem interfejsu**
   - **2.3. Przykład dziedziczenia i polimorfizmu w programie z kalkulatorem**

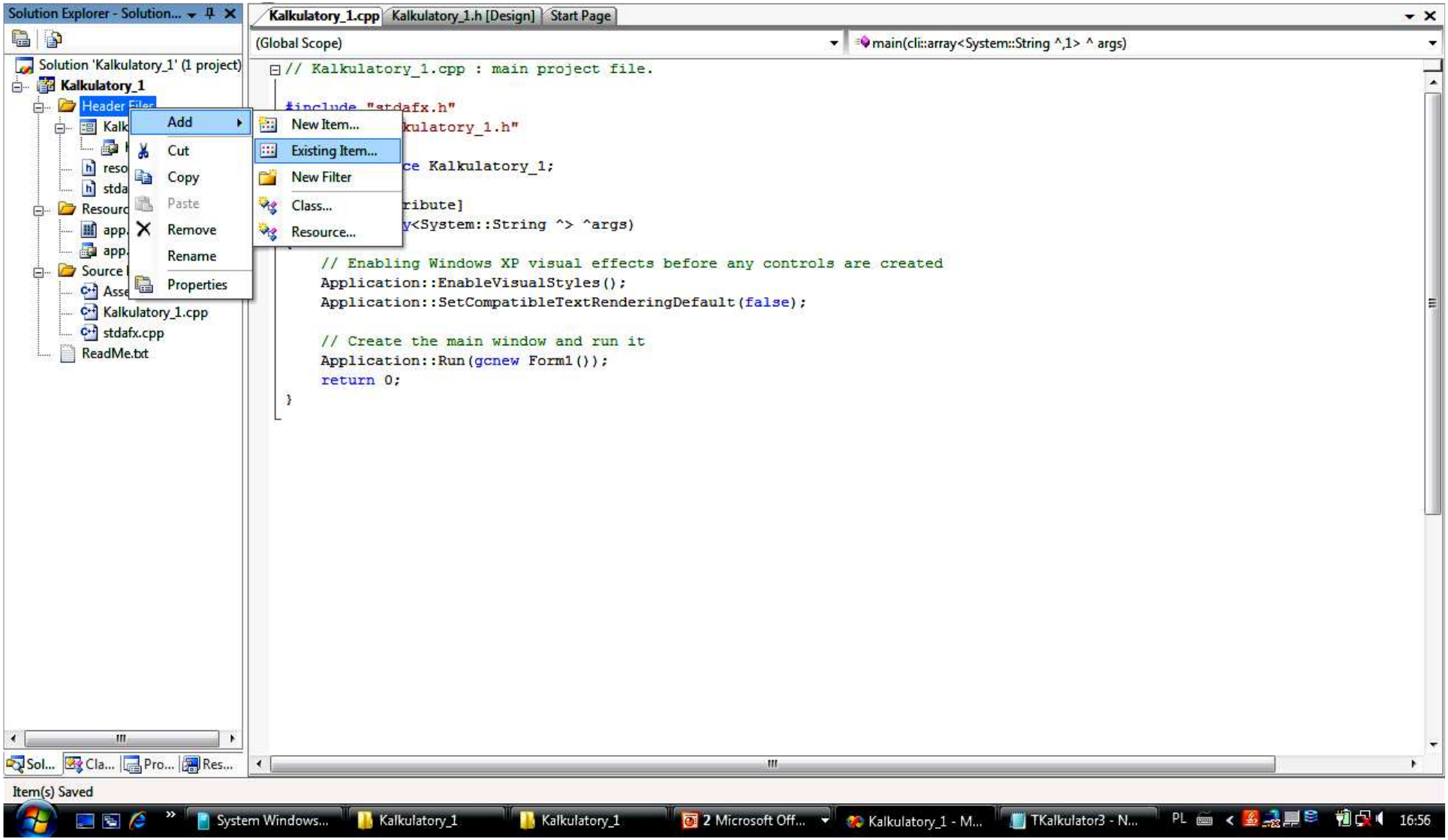
1. Aplikacja wielookienkowa. Zakładanie projektu typu CLR Windows Forms



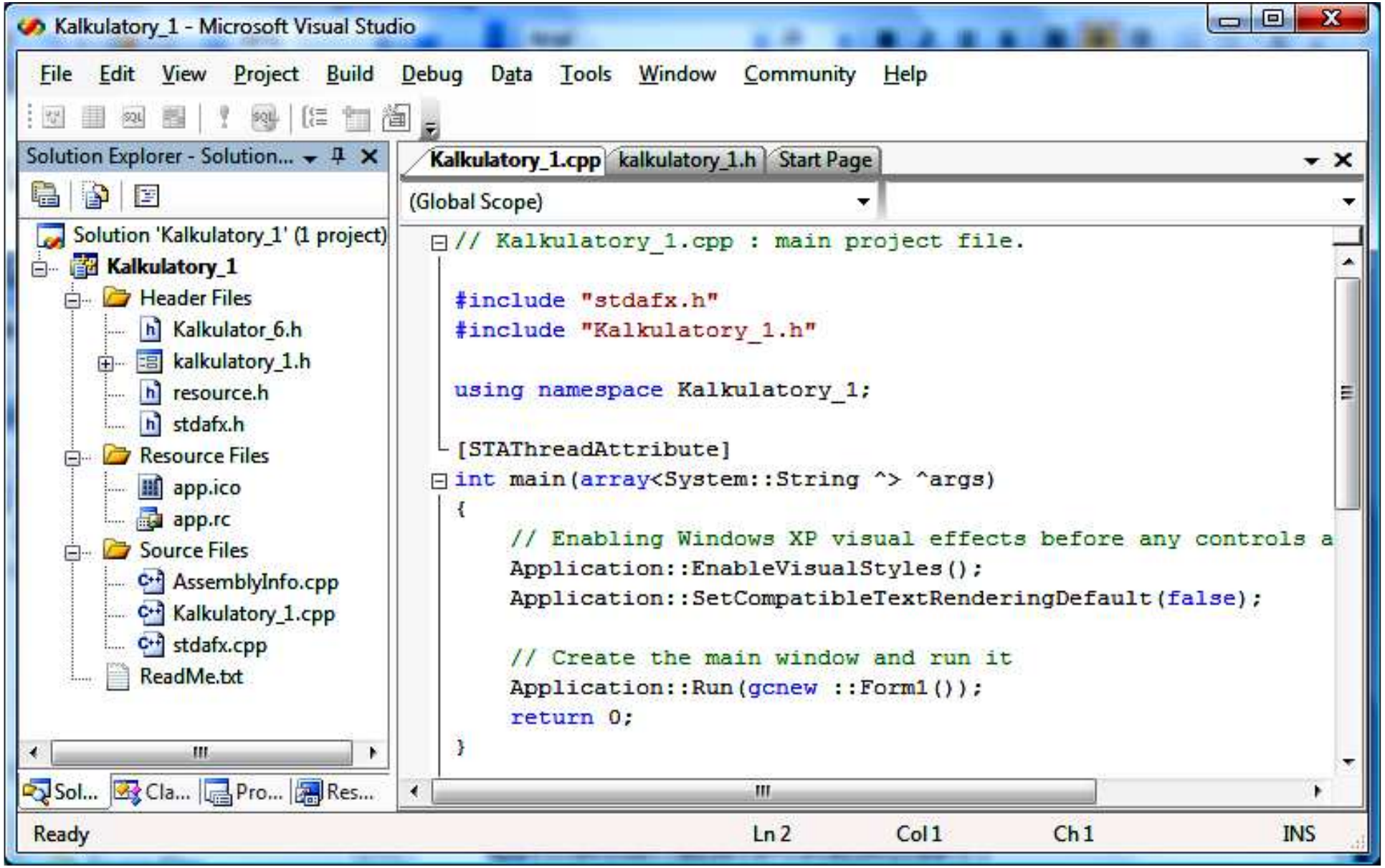
Założenie projektu typu CLR Windows Forms



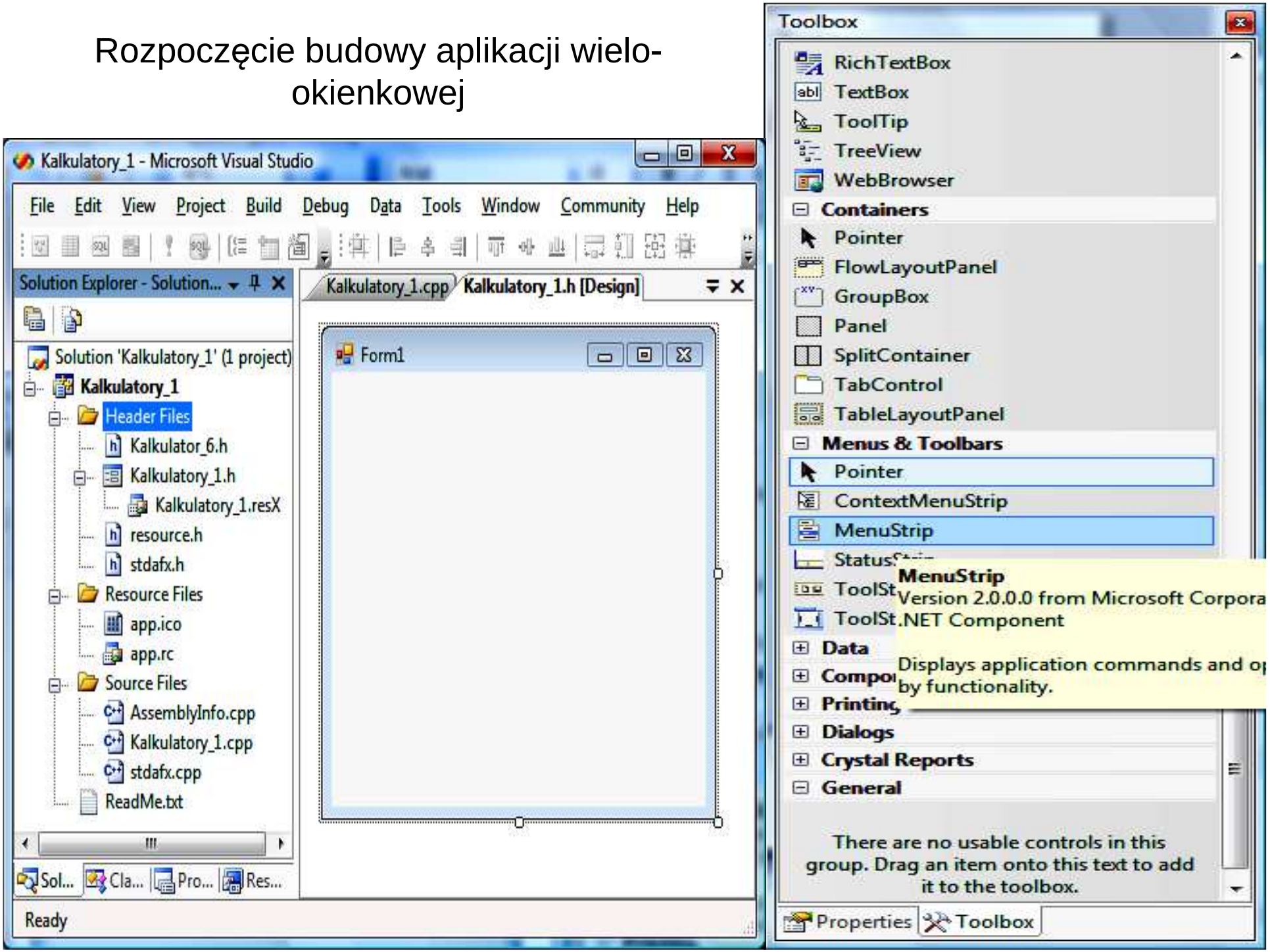
Skopiowanie do katalogu projektu pliku nagłówkowego Kalkulator_6.h z kodem okna Kalkulator3 z projektu Kalkulator_6 (wykład 4) oraz dołączenie go do projektu Kalkulatory_1. Skopiowano również plik nagłówkowy TKalkulator3.h z definicjami klas **TKalkulator_3** oraz **Dzialania_kalkulatora,** używanych w kodzie okna Kalkulator3, dołączany za pomocą dyrektywy **#include** do kodu pliku Kalkulator_6.h.



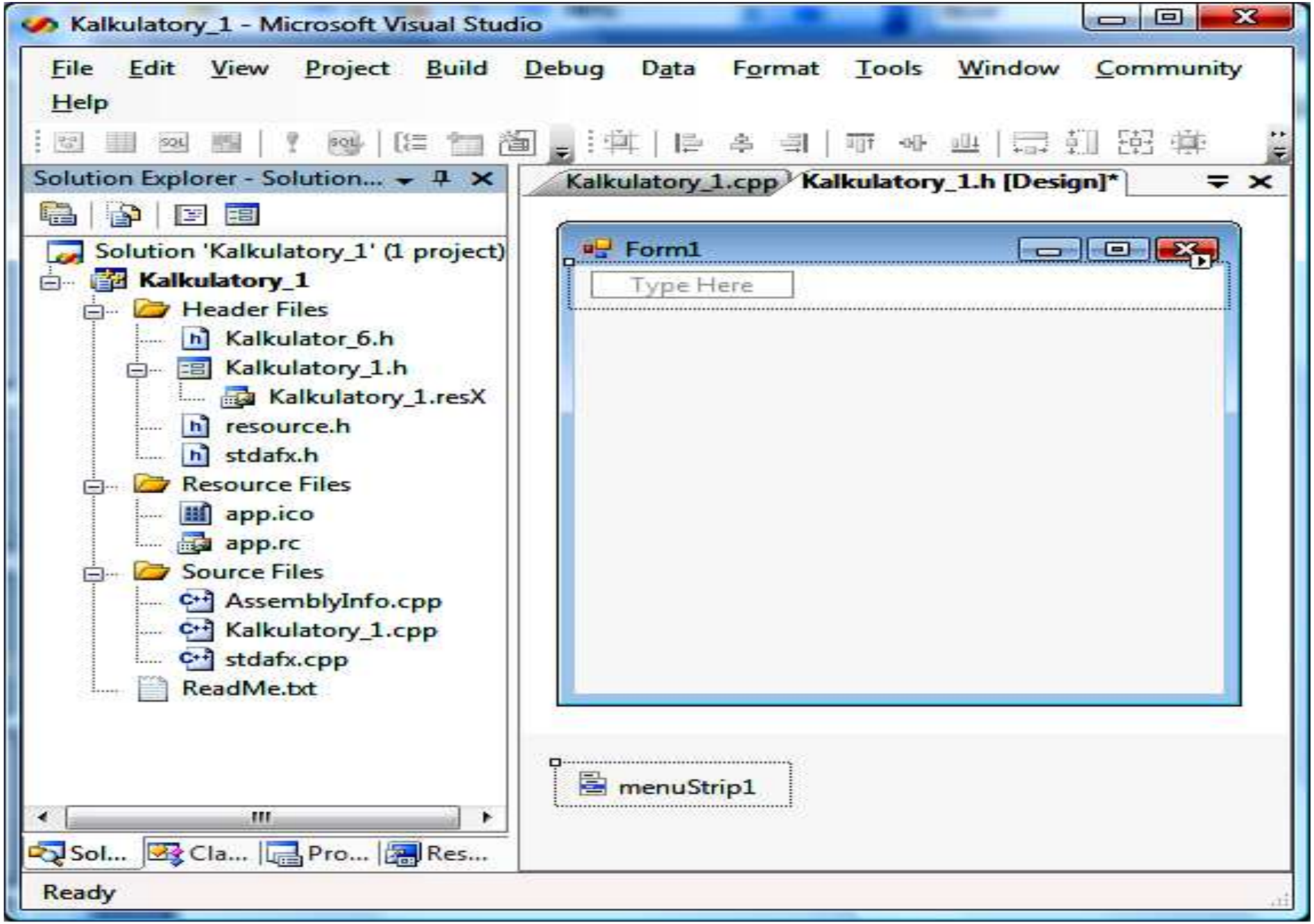
Zmiana nazwy głównego pliku nagłówkowego projektu Form1.h na Kalkulatory_1.h, dołączenie pliku Kalkulator_6.h zostało zaprezentowane na poprzednim slajdzie

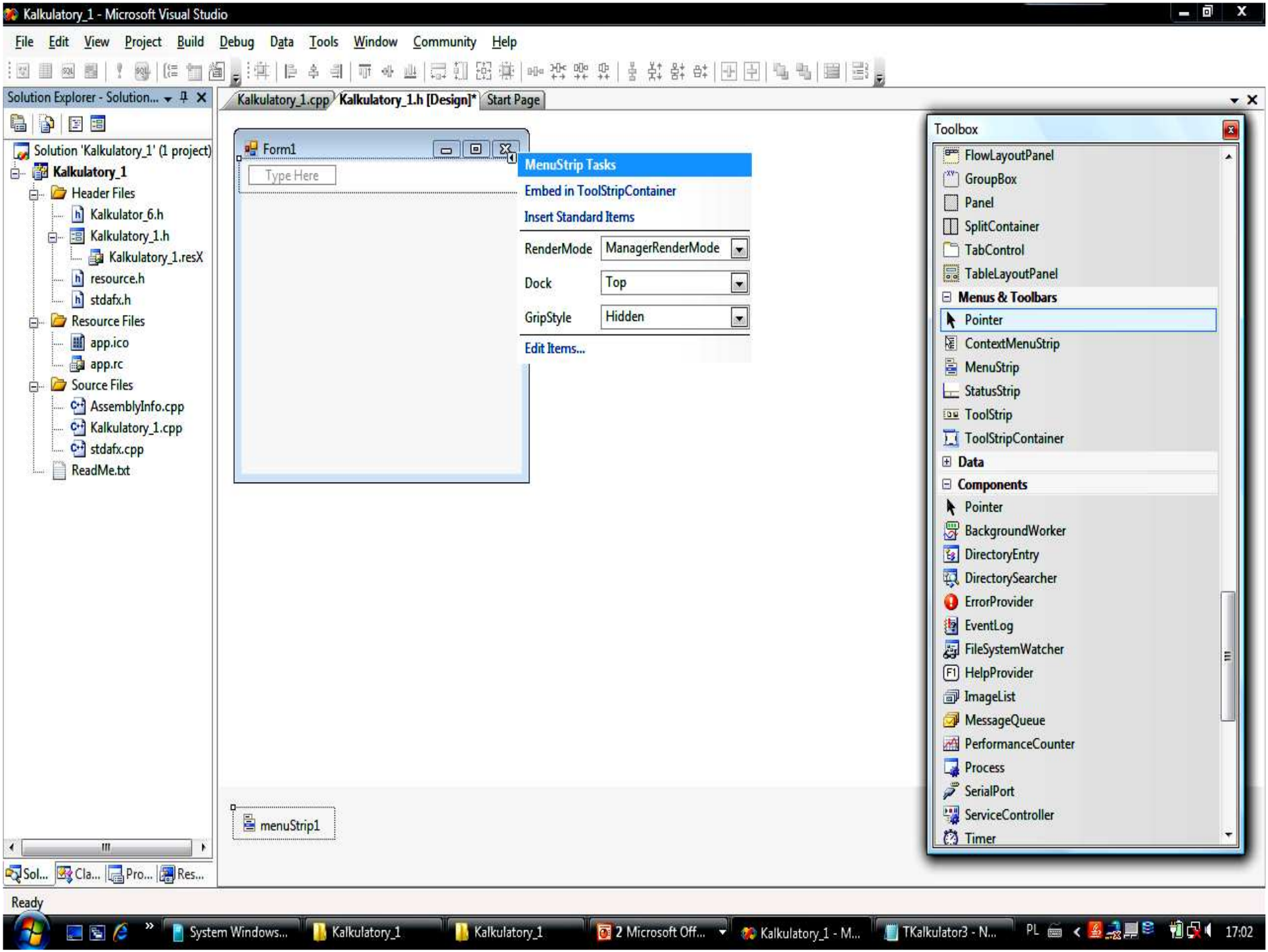


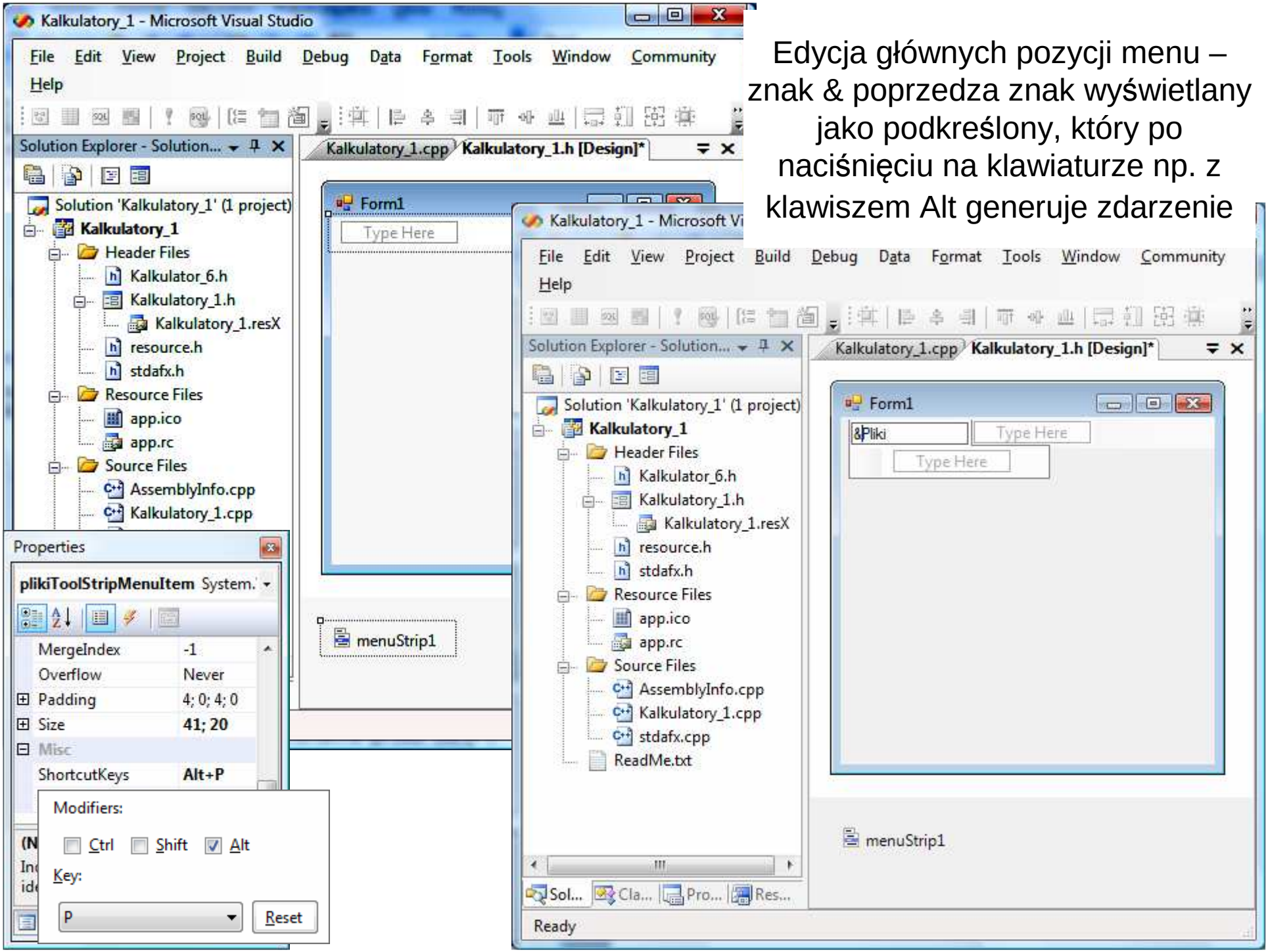
Rozpoczęcie budowy aplikacji wielo-okienkowej



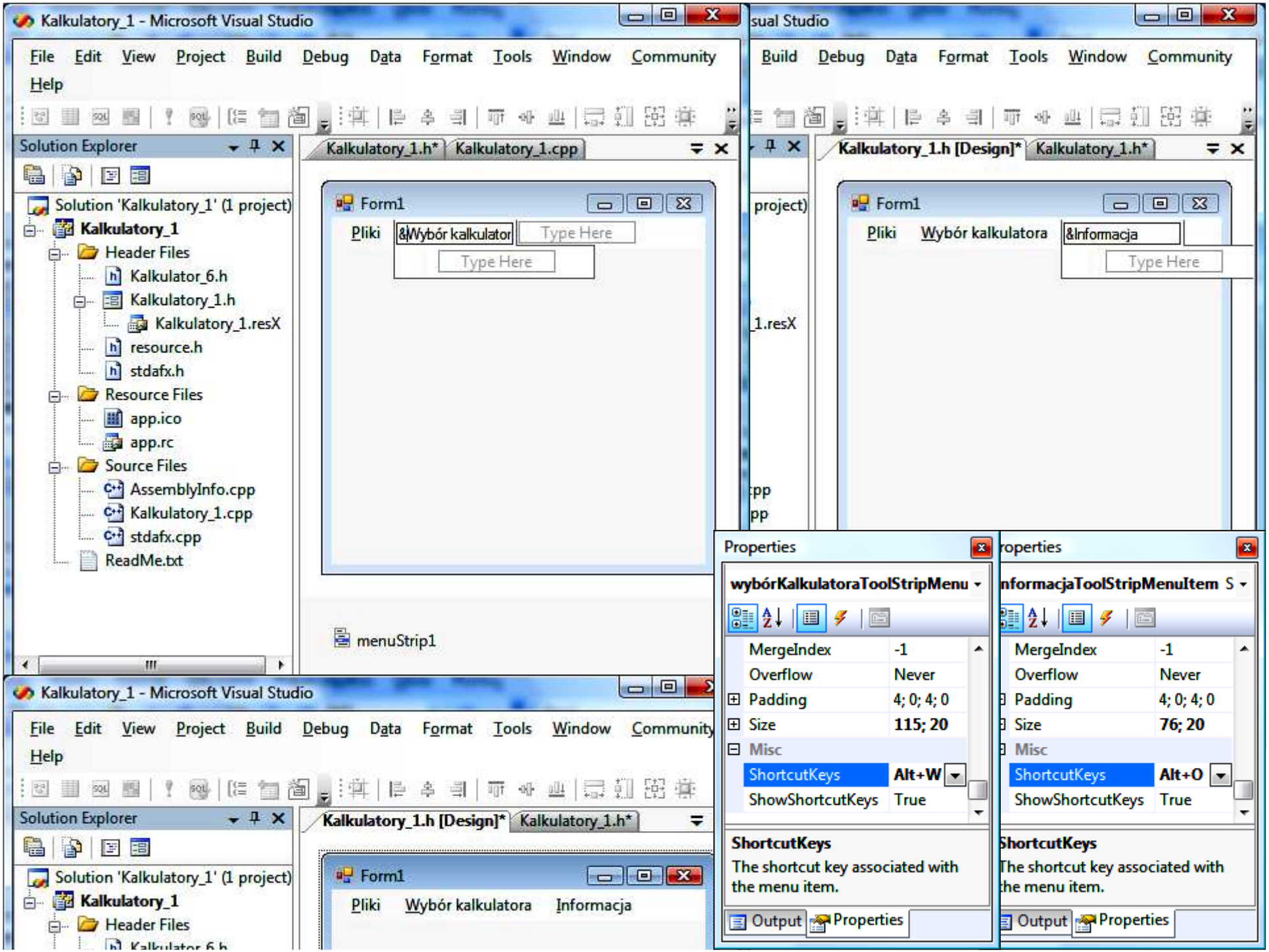
Dołączenie komponentu typu Menu Strip

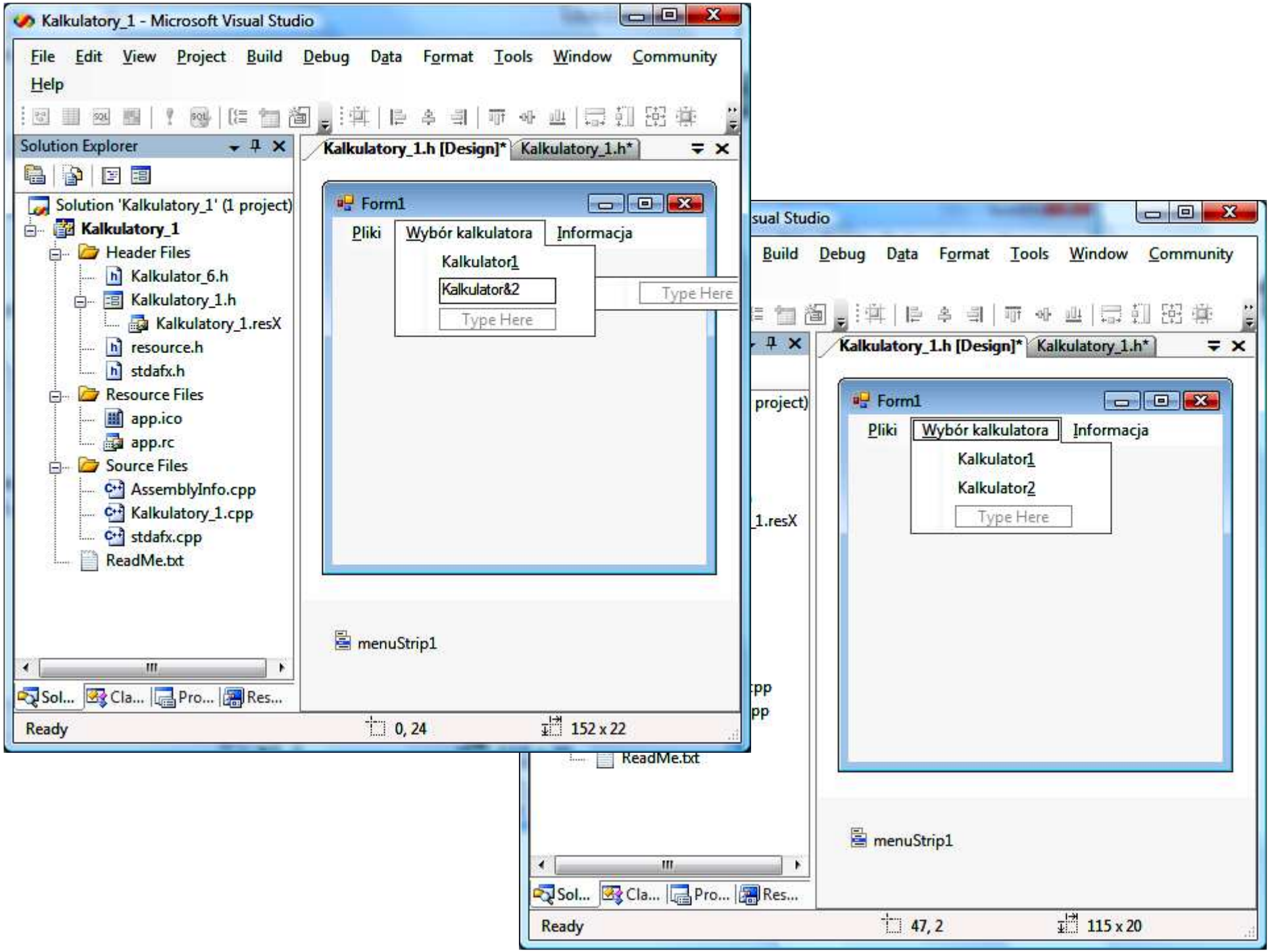


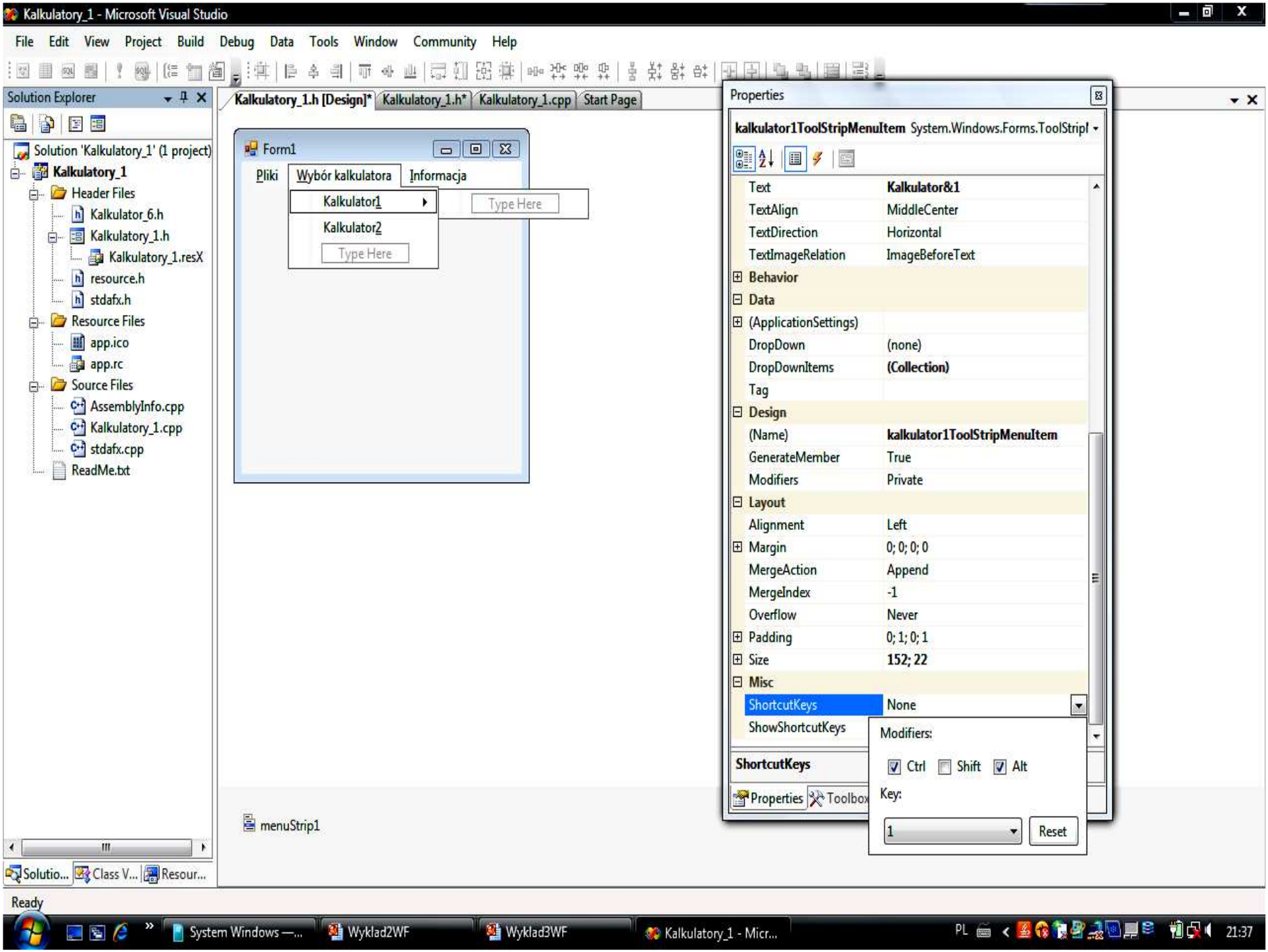

Kalkulatory_1 - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Solution Explorer - Solution...
Solution 'Kalkulatory_1' (1 project)
Kalkulatory_1
Header Files
Kalkulator_6.h
Kalkulatory_1.h
Kalkulatory_1.resX
resource.h
stdafx.h
Resource Files
app.ico
app.rc
Source Files
AssemblyInfo.cpp
Kalkulatory_1.cpp
stdafx.cpp
ReadMe.txt
Kalkulatory_1.cpp
Kalkulatory_1.h [Design]*
Start Page
Form1
Type Here
MenuStrip Tasks
Embed in ToolStripContainer
Insert Standard Items
RenderMode
ManagerRenderMode
Dock
Top
GripStyle
Hidden
Edit Items...
menuStrip1
Toolbox
FlowLayoutPanel
GroupBox
Panel
SplitContainer
TabControl
TableLayoutPanel
Menus & Toolbars
Pointer
ContextMenuStrip
MenuStrip
StatusStrip
ToolStrip
ToolStripContainer
Data
Components
Pointer
BackgroundWorker
DirectoryEntry
DirectorySearcher
ErrorProvider
EventLog
FileSystemWatcher
HelpProvider
ImageList
MessageQueue
PerformanceCounter
Process
SerialPort
ServiceController
Timer
Sol... Cla... Pro... Res...
Ready
System Windows...
Kalkulatory_1
Kalkulatory_1
2 Microsoft Off...
Kalkulatory_1 - M...
TKalkulator3 - N...
PL
17:02

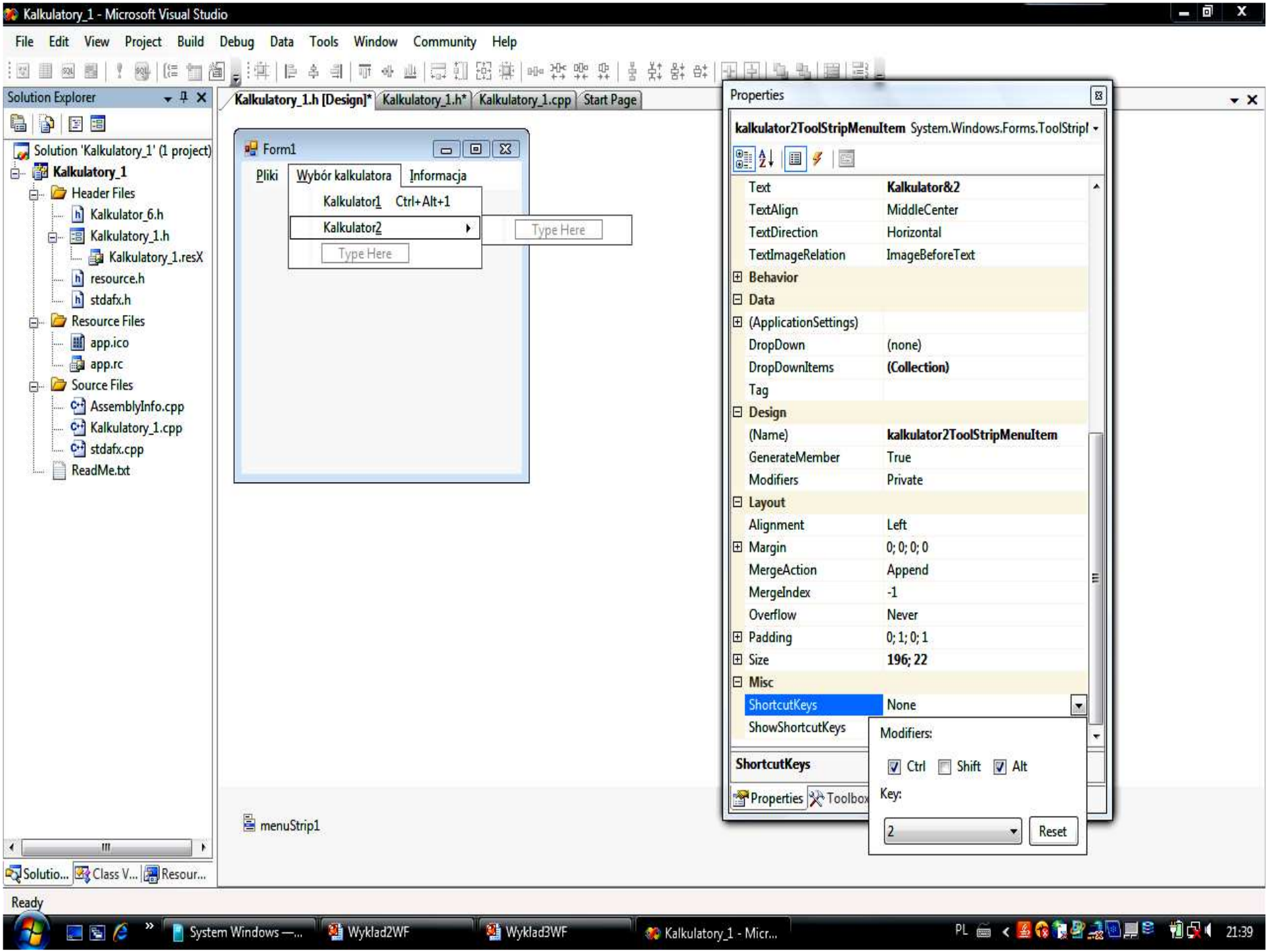


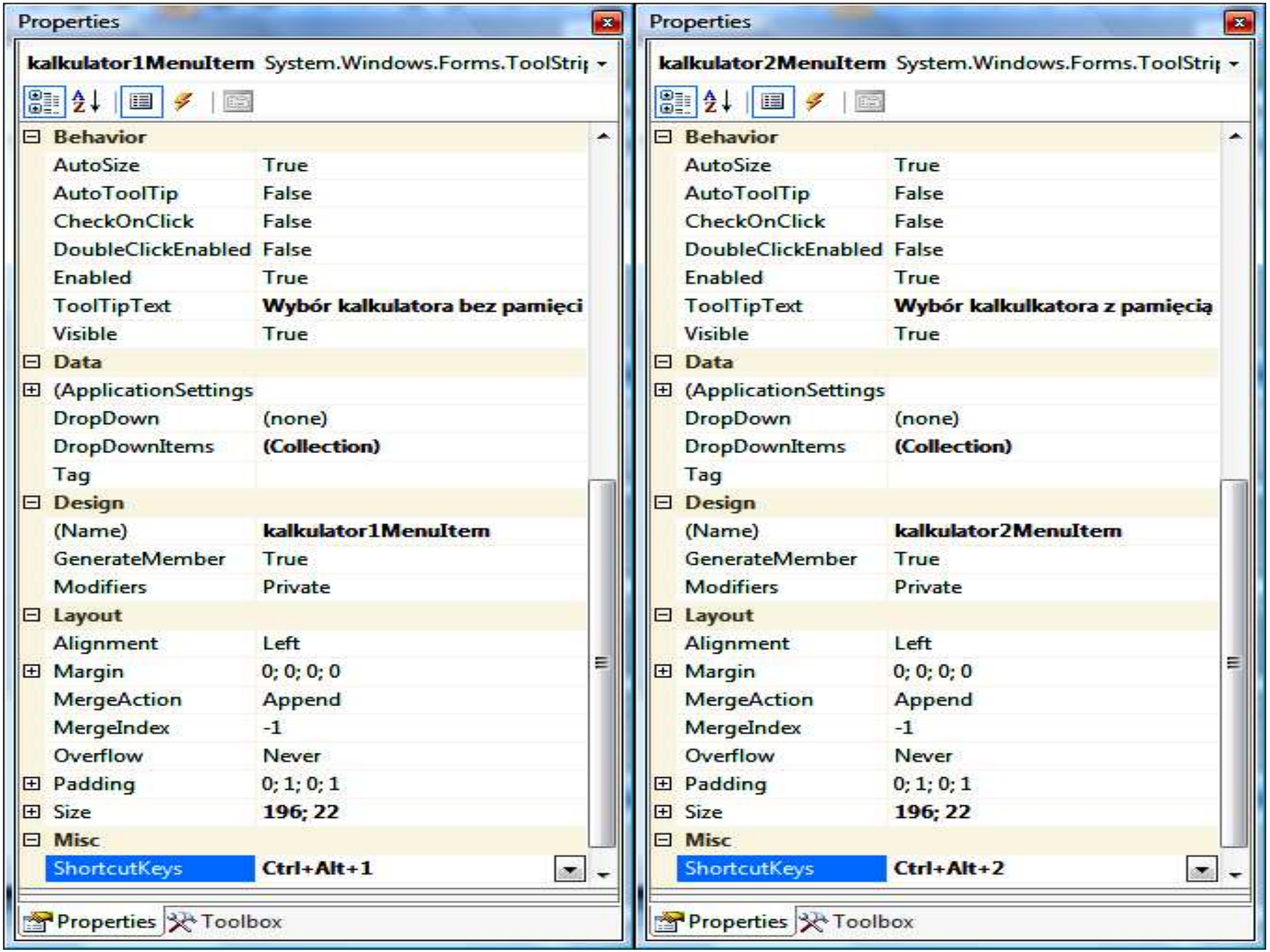
Edycja głównych pozycji menu – znak & poprzedza znak wyświetlany jako podkreślony, który po naciśnięciu na klawiaturze np. z klawiszem Alt generuje zdarzenie

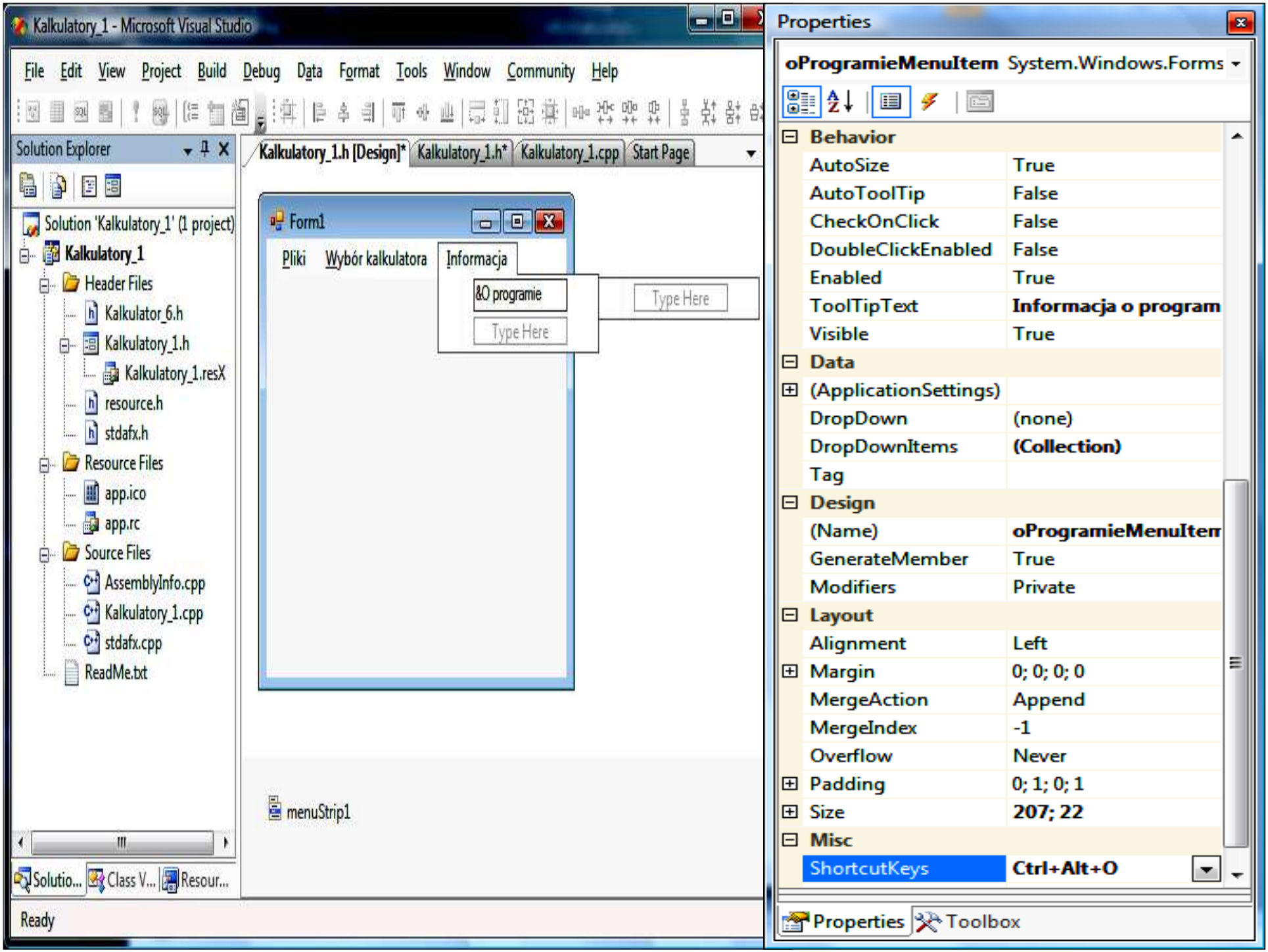

Kalkulatory_1 - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Window Community
Help
Solution Explorer
Solution 'Kalkulatory_1' (1 project)
Kalkulatory_1
Header Files
Kalkulator_6.h
Kalkulatory_1.h
Kalkulatory_1.resX
resource.h
stdafx.h
Resource Files
app.ico
app.rc
Source Files
AssemblyInfo.cpp
Kalkulatory_1.cpp
stdafx.cpp
ReadMe.txt
Kalkulatory_1.h*
Kalkulatory_1.cpp
Form1
Pliki
&Wybór kalkulator
Type Here
menuStrip1
Kalkulatory_1.h [Design]*
Kalkulatory_1.h*
Pliki
Wybór kalkulatora
&Informacja
Properties
wybórKalkulatoraToolStripMenu
MergeIndex -1
Overflow Never
Padding 4; 0; 4; 0
Size 115; 20
Misc
ShortcutKeys Alt+W
ShowShortcutKeys True
ShortcutKeys
The shortcut key associated with the menu item.
Output
Properties
informacjaToolStripMenuItem
Size 76; 20
ShortcutKeys Alt+O
Informacja


Kalkulatory_1 - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Window Community
Help
Solution Explorer
Solution 'Kalkulatory_1' (1 project)
Kalkulatory_1
Header Files
Kalkulator_6.h
Kalkulatory_1.h
Kalkulatory_1.resX
resource.h
stdafx.h
Resource Files
app.ico
app.rc
Source Files
AssemblyInfo.cpp
Kalkulatory_1.cpp
stdafx.cpp
ReadMe.txt
Kalkulatory_1.h [Design]*
Kalkulatory_1.h*
Form1
Pliki
Wybór kalkulatora
Informacja
Kalkulator1
Kalkulator&2
Type Here
Type Here
menuStrip1
Sol...
Cla...
Pro...
Res...
Ready
0, 24
152 x 22
sual Studio
Build Debug Data Format Tools Window Community
project)
_1.resX
pp
pp
ReadMe.txt
Kalkulatory_1.h [Design]*
Kalkulatory_1.h*
Form1
Pliki
Wybór kalkulatora
Informacja
Kalkulator1
Kalkulator2
Type Here
menuStrip1
Sol...
Cla...
Pro...
Res...
Ready
47, 2
115 x 20

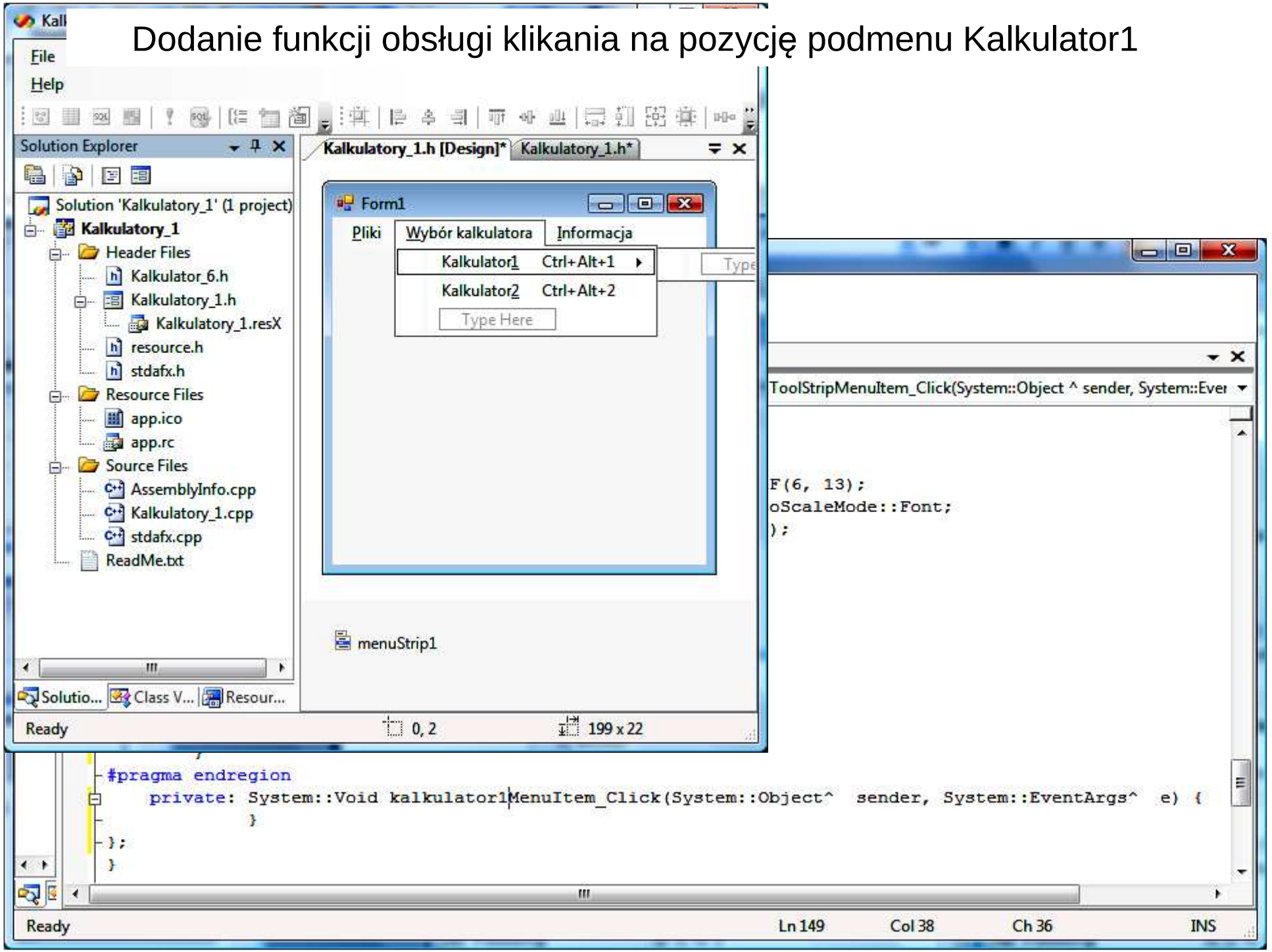

Kalkulatory_1 - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Solution Explorer
Solution 'Kalkulatory_1' (1 project)
Kalkulatory_1
Header Files
Kalkulator_6.h
Kalkulatory_1.h
Kalkulatory_1.resX
resource.h
stdafx.h
Resource Files
app.ico
app.rc
Source Files
AssemblyInfo.cpp
Kalkulatory_1.cpp
stdafx.cpp
ReadMe.txt
Kalkulatory_1.h [Design]*
Kalkulatory_1.h*
Kalkulatory_1.cpp
Start Page
Form1
Pliki
Wybór kalkulatora
Informacja
Kalkulator1
Kalkulator2
Type Here
Type Here
menuStrip1
Properties
kalkulator1ToolStripMenuItem System.Windows.Forms.ToolStripI
Text Kalkulator&1
TextAlign MiddleCenter
TextDirection Horizontal
TextImageRelation ImageBeforeText
Behavior
Data
(ApplicationSettings)
DropDown (none)
DropDownItems (Collection)
Tag
Design
(Name) kalkulator1ToolStripMenuItem
GenerateMember True
Modifiers Private
Layout
Alignment Left
Margin 0; 0; 0; 0
MergeAction Append
MergeIndex -1
Overflow Never
Padding 0; 1; 0; 1
Size 152; 22
Misc
ShortcutKeys None
ShowShortcutKeys
ShortcutKeys
Modifiers:
Ctrl
Shift
Alt
Key:
1
Reset
Properties
Toolbox
Ready
System Windows —...
Wyklad2WF
Wyklad3WF
Kalkulatory_1 - Micr...
PL
21:37


Kalkulatory_1 - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Solution Explorer
Solution 'Kalkulatory_1' (1 project)
Kalkulatory_1
Header Files
Kalkulator_6.h
Kalkulatory_1.h
Kalkulatory_1.resX
resource.h
stdafx.h
Resource Files
app.ico
app.rc
Source Files
AssemblyInfo.cpp
Kalkulatory_1.cpp
stdafx.cpp
ReadMe.txt
Kalkulatory_1.h [Design]*
Kalkulatory_1.h*
Kalkulatory_1.cpp
Start Page
Form1
Pliki
Wybór kalkulatora
Informacja
Kalkulator1 Ctrl+Alt+1
Kalkulator2
Type Here
Type Here
menuStrip1
Properties
kalkulator2ToolStripMenuItem System.Windows.Forms.ToolStripI
Text Kalkulator&2
TextAlign MiddleCenter
TextDirection Horizontal
TextImageRelation ImageBeforeText
Behavior
Data
(ApplicationSettings)
DropDown (none)
DropDownItems (Collection)
Tag
Design
(Name) kalkulator2ToolStripMenuItem
GenerateMember True
Modifiers Private
Layout
Alignment Left
Margin 0; 0; 0; 0
MergeAction Append
MergeIndex -1
Overflow Never
Padding 0; 1; 0; 1
Size 196; 22
Misc
ShortcutKeys None
ShowShortcutKeys
ShortcutKeys
Modifiers:
Ctrl Shift Alt
Key:
2
Reset
Properties Toolbox
Ready
System Windows —...
Wyklad2WF
Wyklad3WF
Kalkulatory_1 - Micr...
PL
21:39

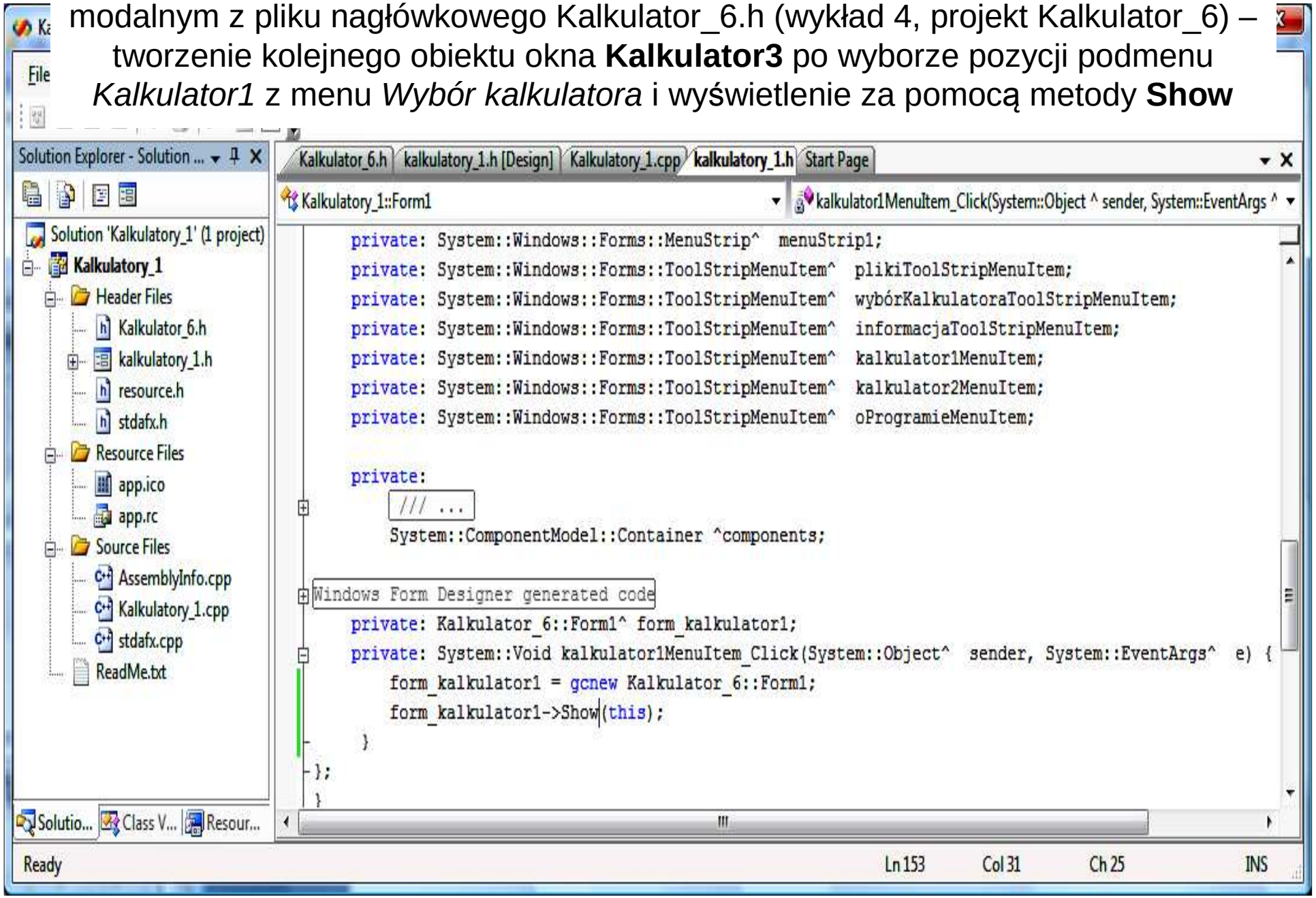

Properties
kalkulator1MenuItem System.Windows.Forms.ToolStrip
Behavior
AutoSize True
AutoToolTip False
CheckOnClick False
DoubleClickEnabled False
Enabled True
ToolTipText Wybór kalkulatora bez pamięci
Visible True
Data
(ApplicationSettings
DropDown (none)
DropDownItems (Collection)
Tag
Design
(Name) kalkulator1MenuItem
GenerateMember True
Modifiers Private
Layout
Alignment Left
Margin 0; 0; 0; 0
MergeAction Append
MergeIndex -1
Overflow Never
Padding 0; 1; 0; 1
Size 196; 22
Misc
ShortcutKeys Ctrl+Alt+1
Properties
Toolbox
Properties
kalkulator2MenuItem System.Windows.Forms.ToolStrip
Behavior
AutoSize True
AutoToolTip False
CheckOnClick False
DoubleClickEnabled False
Enabled True
ToolTipText Wybór kalkulkatora z pamięcią
Visible True
Data
(ApplicationSettings
DropDown (none)
DropDownItems (Collection)
Tag
Design
(Name) kalkulator2MenuItem
GenerateMember True
Modifiers Private
Layout
Alignment Left
Margin 0; 0; 0; 0
MergeAction Append
MergeIndex -1
Overflow Never
Padding 0; 1; 0; 1
Size 196; 22
Misc
ShortcutKeys Ctrl+Alt+2
Properties
Toolbox


Kalkulatory_1 - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Window Community Help
Solution Explorer
Solution 'Kalkulatory_1' (1 project)
Kalkulatory_1
Header Files
Kalkulator_6.h
Kalkulatory_1.h
Kalkulatory_1.resX
resource.h
stdafx.h
Resource Files
app.ico
app.rc
Source Files
AssemblyInfo.cpp
Kalkulatory_1.cpp
stdafx.cpp
ReadMe.txt
Kalkulatory_1.h [Design]*
Kalkulatory_1.h*
Kalkulatory_1.cpp
Start Page
Form1
Pliki
Wybór kalkulatora
Informacja
&O programie
Type Here
Type Here
menuStrip1
Solutio...
Class V...
Resour...
Ready
Properties
oProgramieMenuItem System.Windows.Forms
Behavior
AutoSize True
AutoToolTip False
CheckOnClick False
DoubleClickEnabled False
Enabled True
ToolTipText Informacja o program
Visible True
Data
(ApplicationSettings)
DropDown (none)
DropDownItems (Collection)
Tag
Design
(Name) oProgramieMenuItem
GenerateMember True
Modifiers Private
Layout
Alignment Left
Margin 0; 0; 0; 0
MergeAction Append
MergeIndex -1
Overflow Never
Padding 0; 1; 0; 1
Size 207; 22
Misc
ShortcutKeys Ctrl+Alt+O
Properties
Toolbox

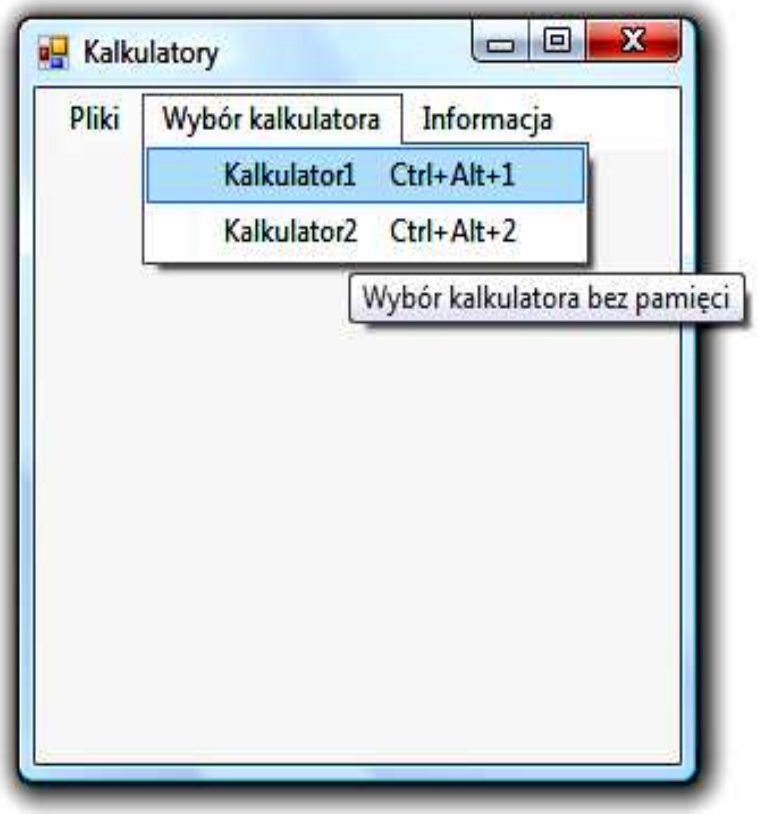
Dodanie funkcji obsługi klikania na pozycję podmenu Kalkulator1



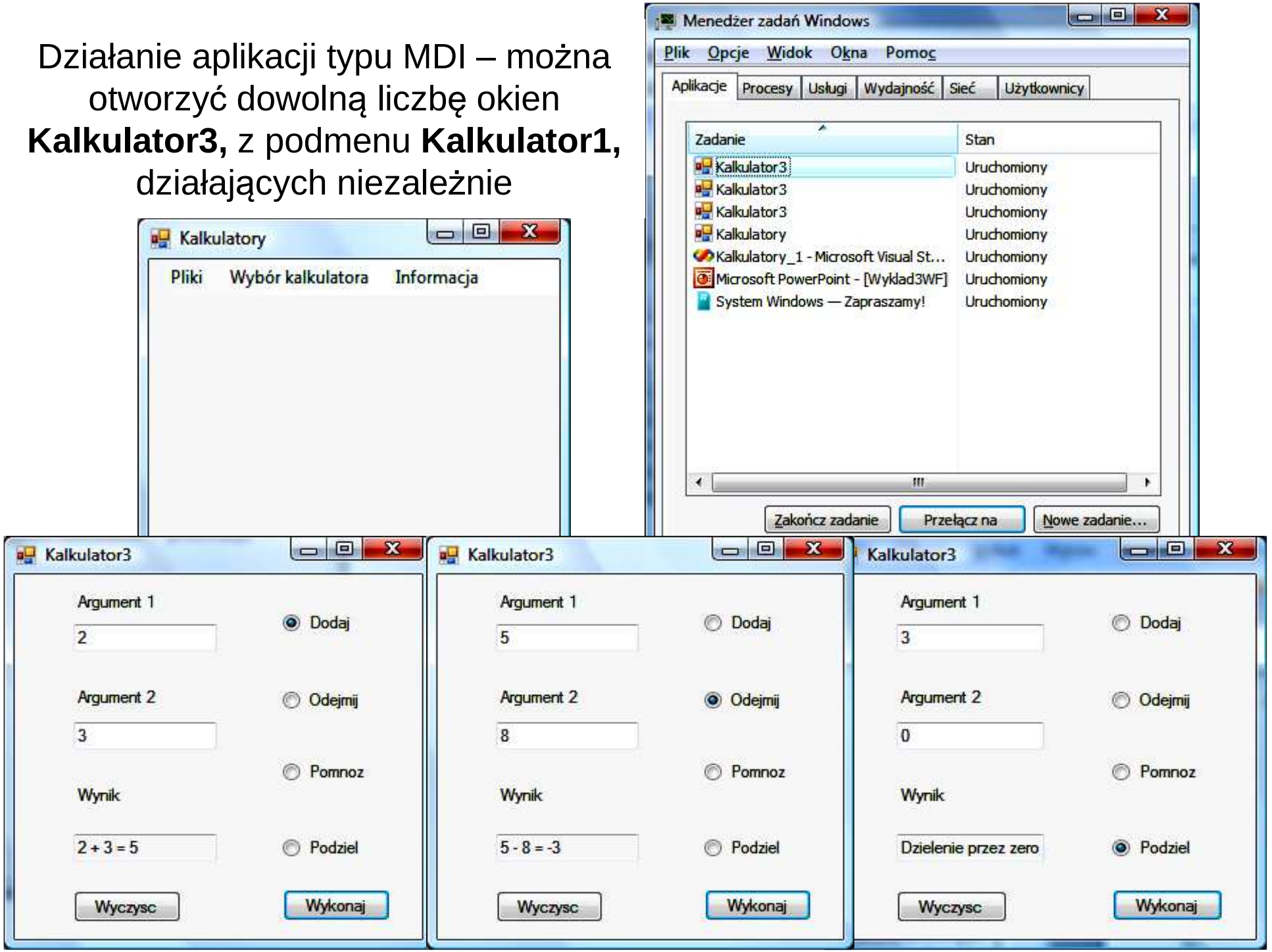
1.1. Aplikacja typu MDI - obsługa zdarzenia wywołania kalkulatora w trybie nie modalnym z pliku nagłówkowego Kalkulator_6.h (wykład 4, projekt Kalkulator_6) – tworzenie kolejnego obiektu okna **Kalkulator3** po wyborze pozycji podmenu *Kalkulator1* z menu *Wybór kalkulatora* i wyświetlenie za pomocą metody **Show**



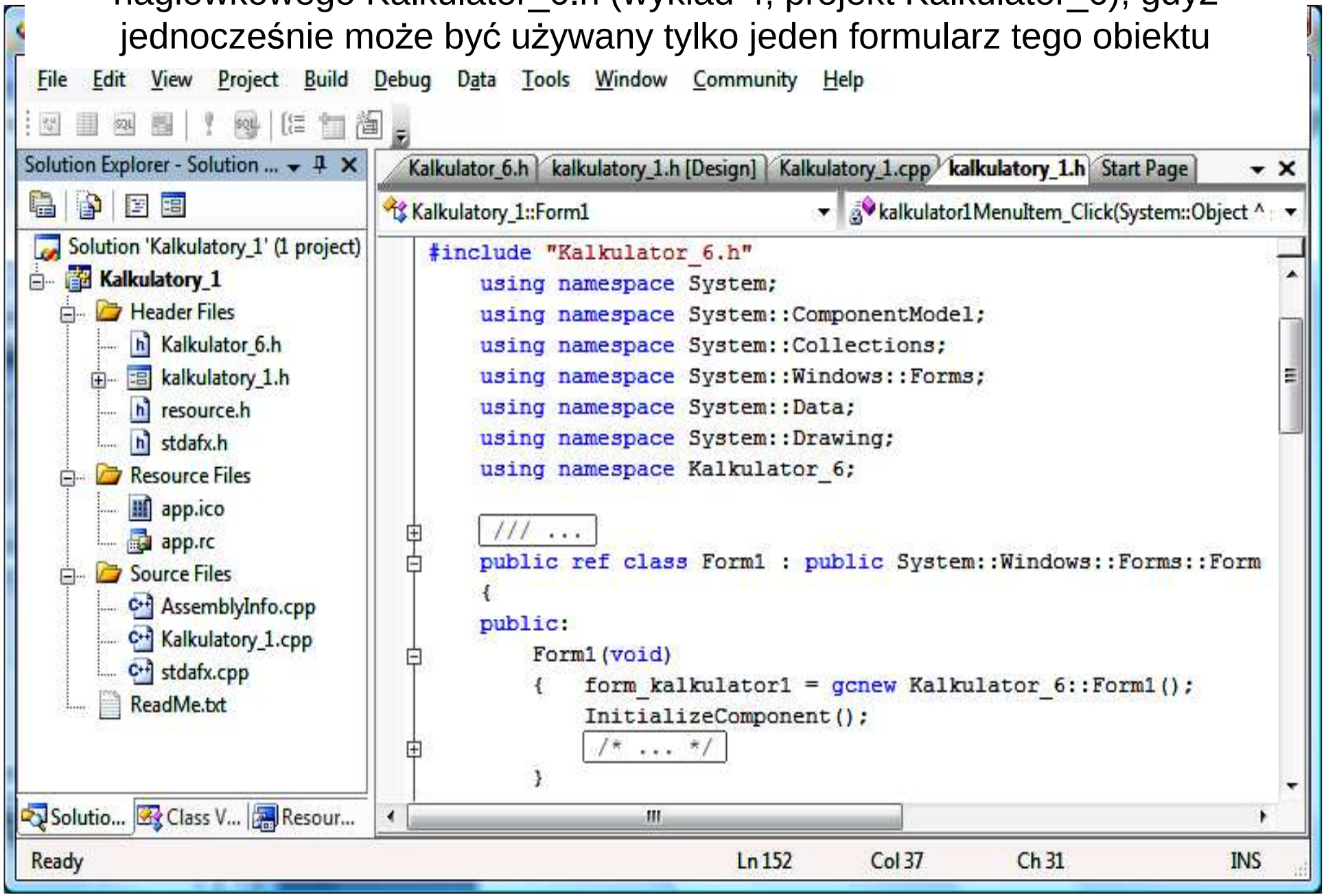

Kalkulatory
Pliki
Wybór kalkulatora
Informacja


Kalkulatory
Pliki
Wybór kalkulatora
Informacja
Kalkulator1 Ctrl+Alt+1
Kalkulator2 Ctrl+Alt+2
Wybór kalkulatora bez pamięci

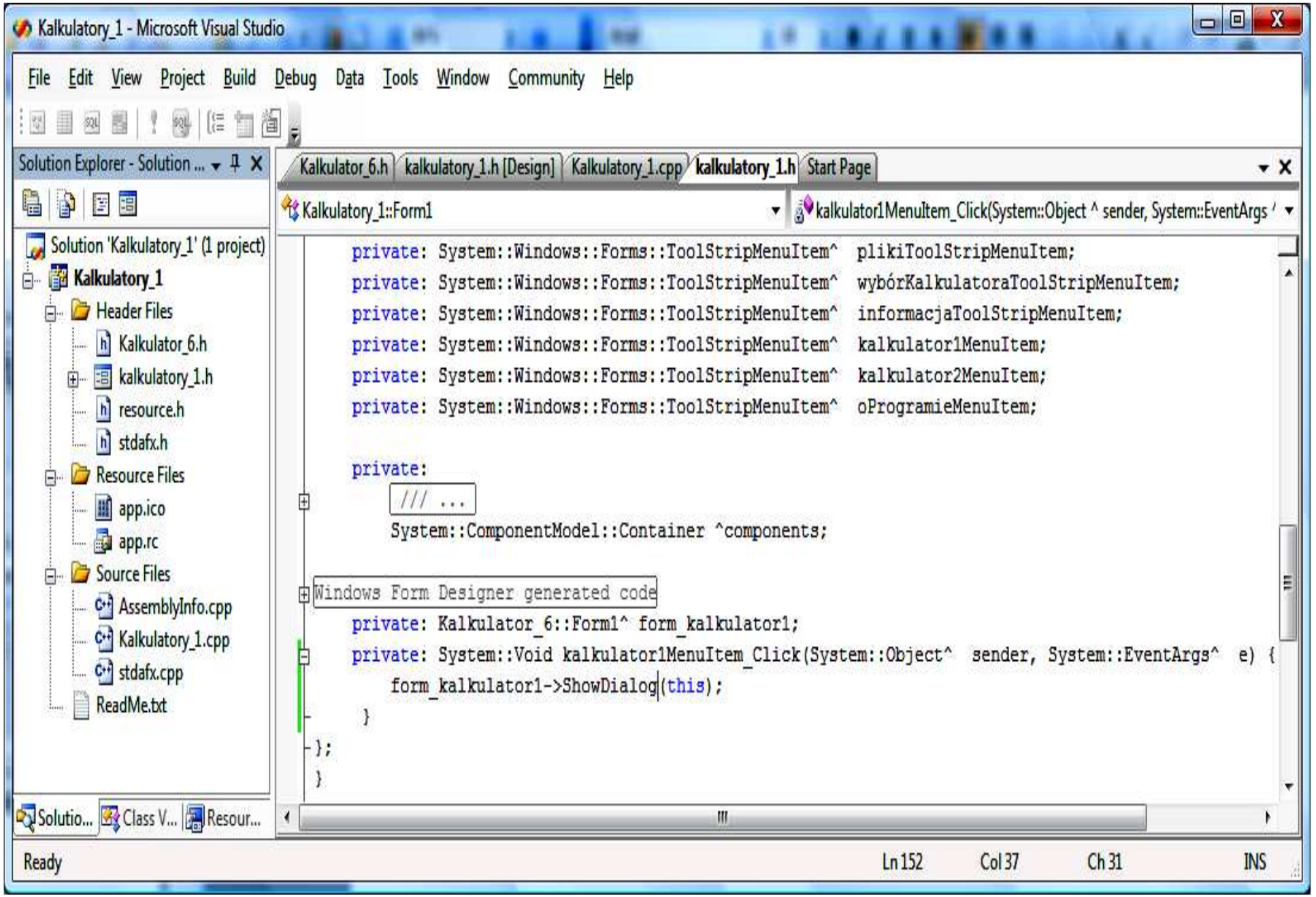
Działanie aplikacji typu MDI – można otworzyć dowolną liczbę okien **Kalkulator3,** z podmenu **Kalkulator1,** działających niezależnie

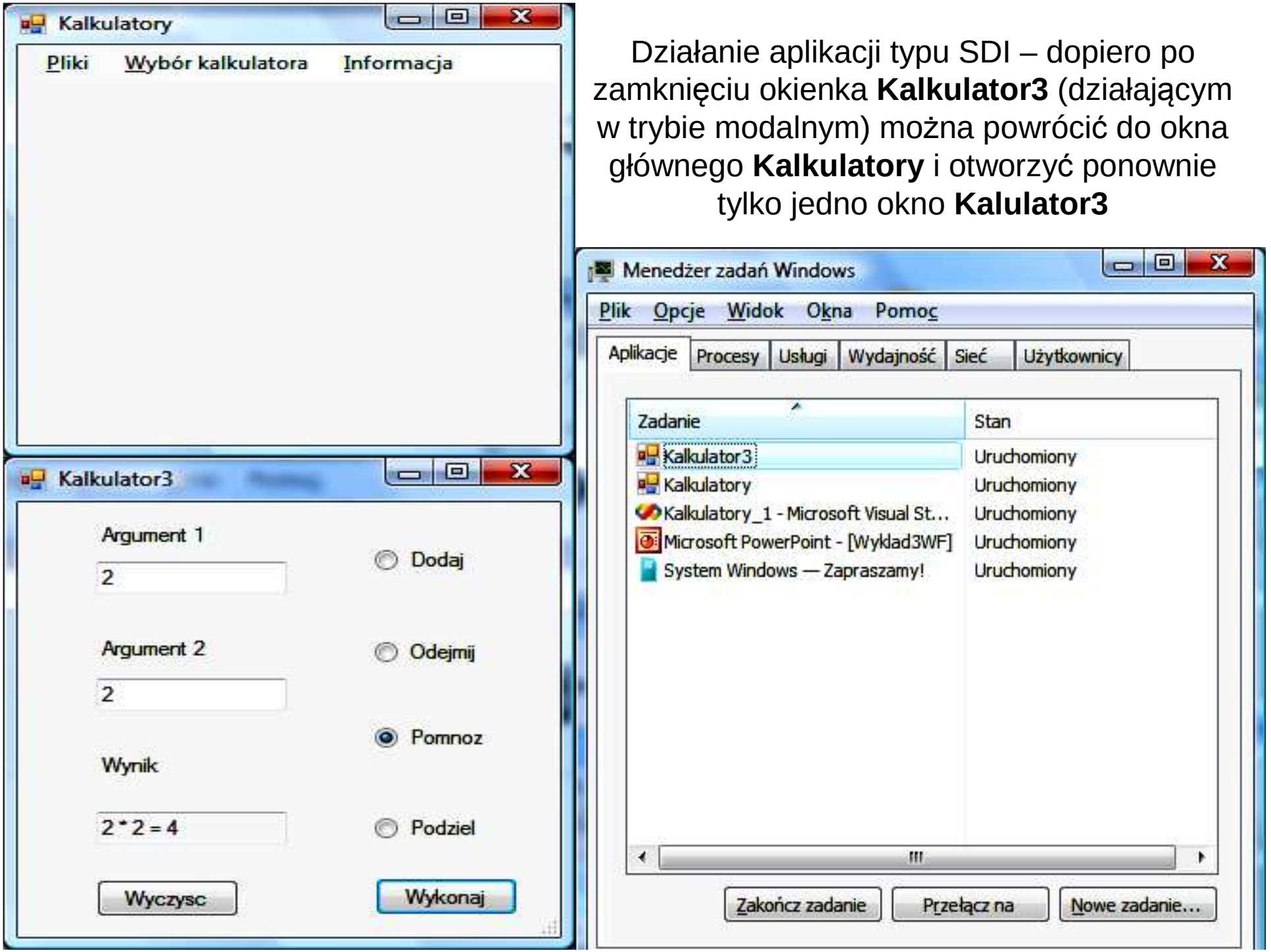


1.2. Aplikacja typu SDI - utworzenie jednego obiektu kalkulatora z pliku nagłówkowego Kalkulator_6.h (wykład 4, projekt Kalkulator_6), gdyż jednocześnie może być używany tylko jeden formularz tego obiektu



Obsługa zdarzenia wyświetlenia formularza kalkulatora w trybie modalnym za pomocą metody ShowDialog- uzyskano aplikację typu SDI





Działanie aplikacji typu SDI – dopiero po zamknięciu okienka **Kalkulator3** (działającym w trybie modalnym) można powrócić do okna głównego **Kalkulatory** i otworzyć ponownie tylko jedno okno **Kalulator3**

2. Dziedziczenie (1)

- Wszystkie klasy typów **value** i **ref** dziedziczą od klasy **System::Object,** jednak nie można tworzyć klas pochodnych od klas typu **value – jedynie** klasy **typu ref** mogą posiadać klasy pochodne
- Każda klasa dziedziczy od klasy **System::Object** metody typu **virtual**:

String^ ToString()	Zwraca po przedefiniowaniu łańcuchowa reprezentację pól klasy (o sposobie prezentacji decyduje programista)
bool REquals(Object^ obj)	Zwraca po przedefiniowaniu wynik **true** lub **false** jako wynik porównania pól obiektu obj z polami obiektu, dla którego wywołano metodę (o sposobie porównania decyduje programista)
int GetHashCode()	Zwraca kod mieszający, używany jako numer miejsca np. w kolekcji, w którym przechowywa-ny jest dany obiekt

- W klasach typów **value** i **ref** można przedefiniować podane wyżej metody wirtualne,
- Klasy typu **value** pozwalają na pakowanie (**boxing**) i odpakowanie (**unboxing**) wartości typów fundamentalnych (**int, double**) za pomocą uchwytu typu System::Object^ - stąd wartości typów fundamentalnych mogą zachowywać się w pewnych wypadkach jak obiekty, a w innych jako wartości, które biorą udział w obliczeniach

Właściwości klas referencyjnych (2)

- Tylko klasy referencyjne mogą być typami klas pochodnych
- Klasa bazowa klasy referencyjnej jest zawsze publiczna
- Metoda bez definicji ciała jest metodą abstrakcyjną i musi być zadeklarowana jako **abstract**
- Klasa, która zawiera metody zadeklarowane jako **abstract** musi być zadeklarowana jako **abstract** (słowo **abstract** umieszczone po nazwie klasy)
- Metoda wirtualna musi być oznaczona przez słowo **virtual** umieszczone z lewej strony nagłówka metody
- Metodę wirtualną można przedefiniować w klasie pochodnej – zawsze należy powtórzyć słowo **virtual** i zawsze należy dodać z prawej strony nagłówka metody słowo **override**
- Klasa interfejsowa deklaruje zestaw metod zaimplementowanych przez klasy typu **ref** i **value**. Jest deklarowana jako **interface class** lub **interface struct**.
- Klasy interfejsowe mogą dziedziczyć po innych interfejsach. Metody deklarowane w interfejsie są zawsze abstrakcyjne i wirtualne

Specyfikatory dostępu do klas i interfejsów oraz ich składowych (3)

- Program napisany w C++/CLI rezyduje zawsze w jednej lub kilku asemblacjach
- Specyfikatory widoczności decydują o dostępie klasy i jej składowych
  - Specyfikator dostępu interfejsu i klasy**: private** lub **public**

    np.. **public interface class** Figura { }; // dostęp do interfejsu z asemblacji //zewnętrznej. Klasy prywatne są widoczne wewnątrz asemblacji //wewnętrznej. Dostęp ten jest domyślny.
  - Specyfikatory dostępu do składowych klas i interfejsów (domyślnie składowe interfejsu są publiczne)

public ref class Figura //klasa widoczna poza asemblacją

{

public: //składowe dostępne do klas wewnątrz i na zewnątrz asemblacji nadrzędnej

internal: // składowe dostępne do klas wewnątrz i na zewnątrz asemblacji nadrzędnej

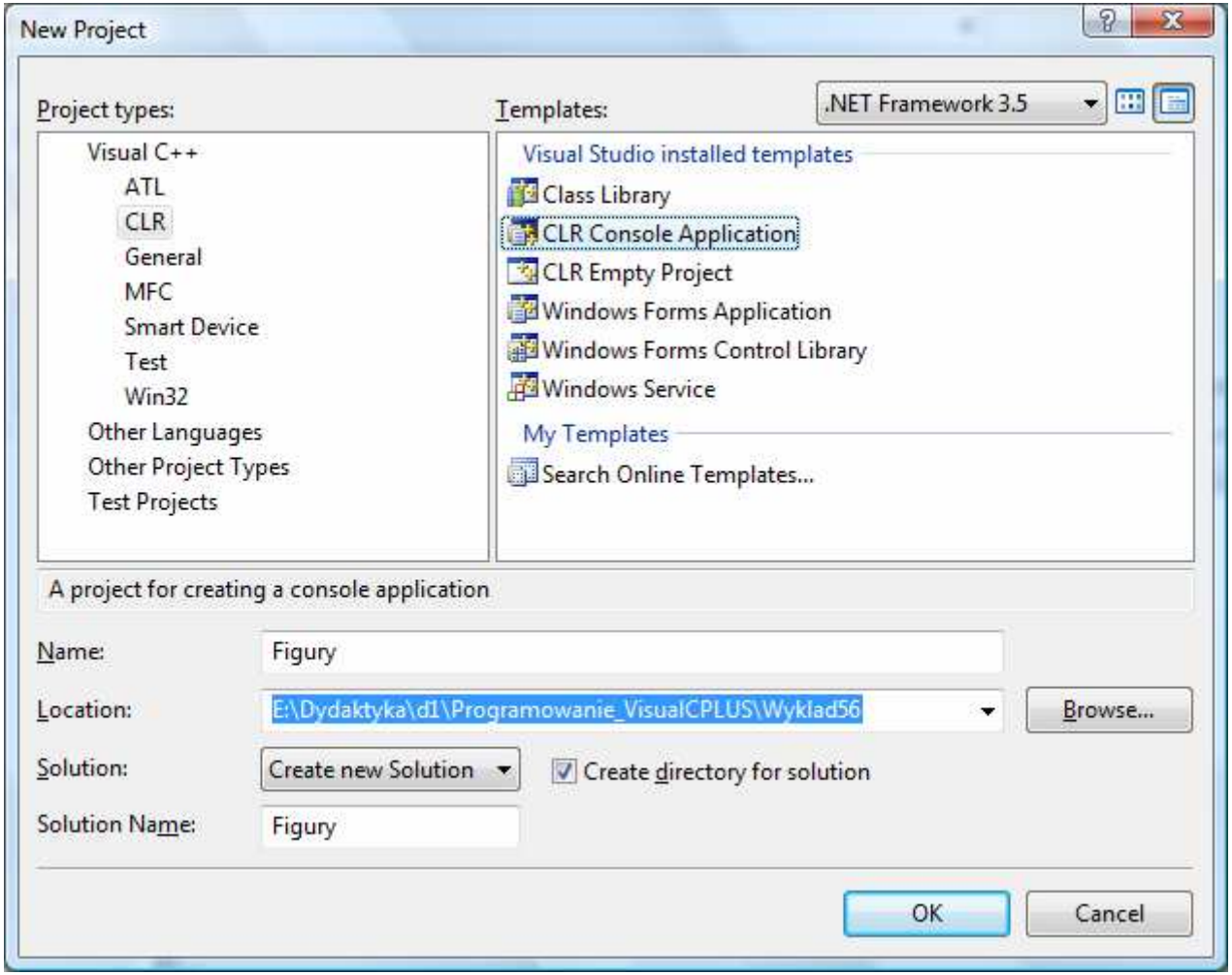
public protected: //składowe dostępne dla typów pochodnych klasy Figura

// poza asemblacją nadrzędną oraz dla klas wewnątrz asemblacji nadrzędnej

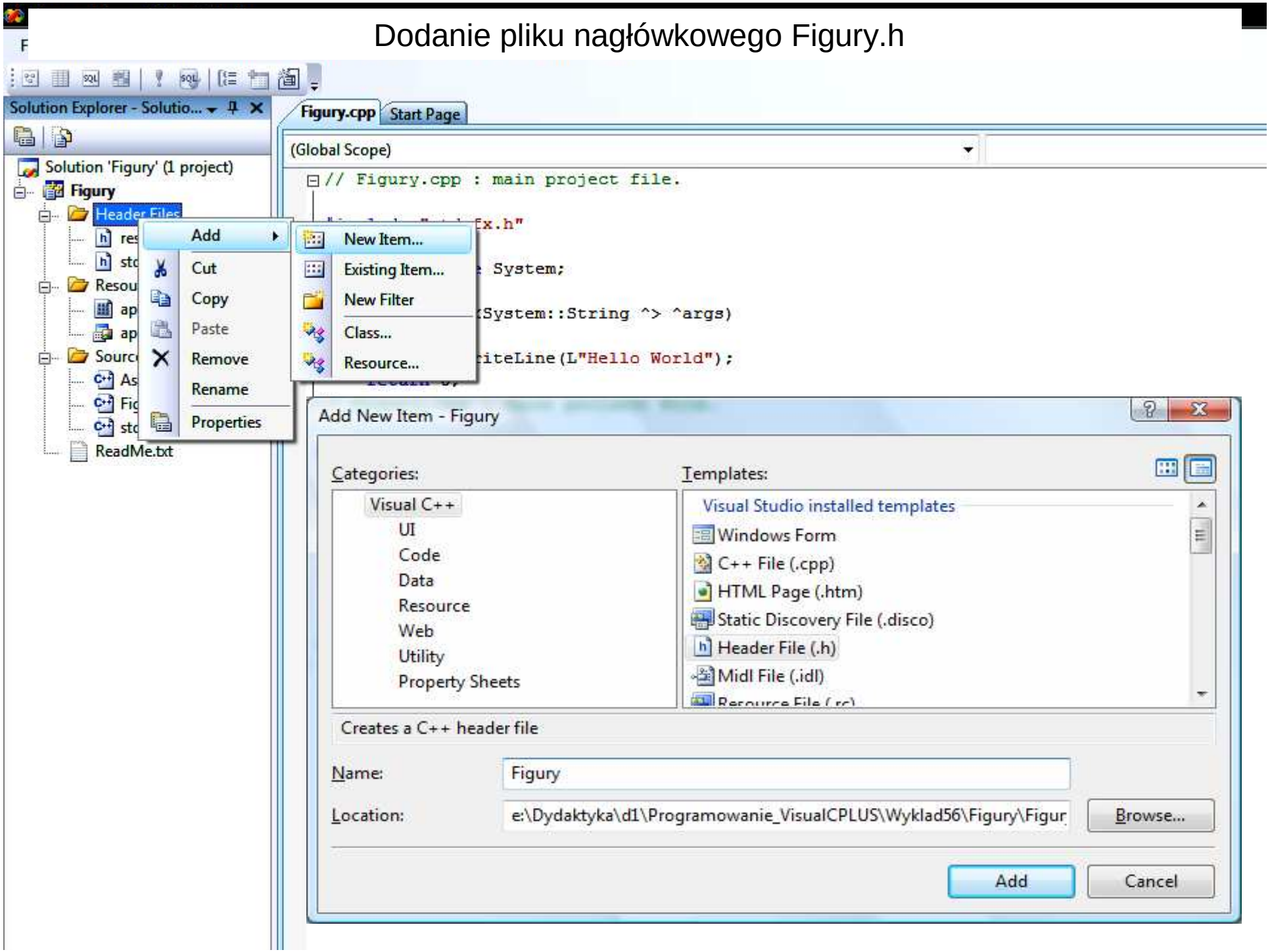
private protected: //składowe dostępne dla typów pochodnych klasy Figura //wewnątrz asemblacji nadrzędnej

};

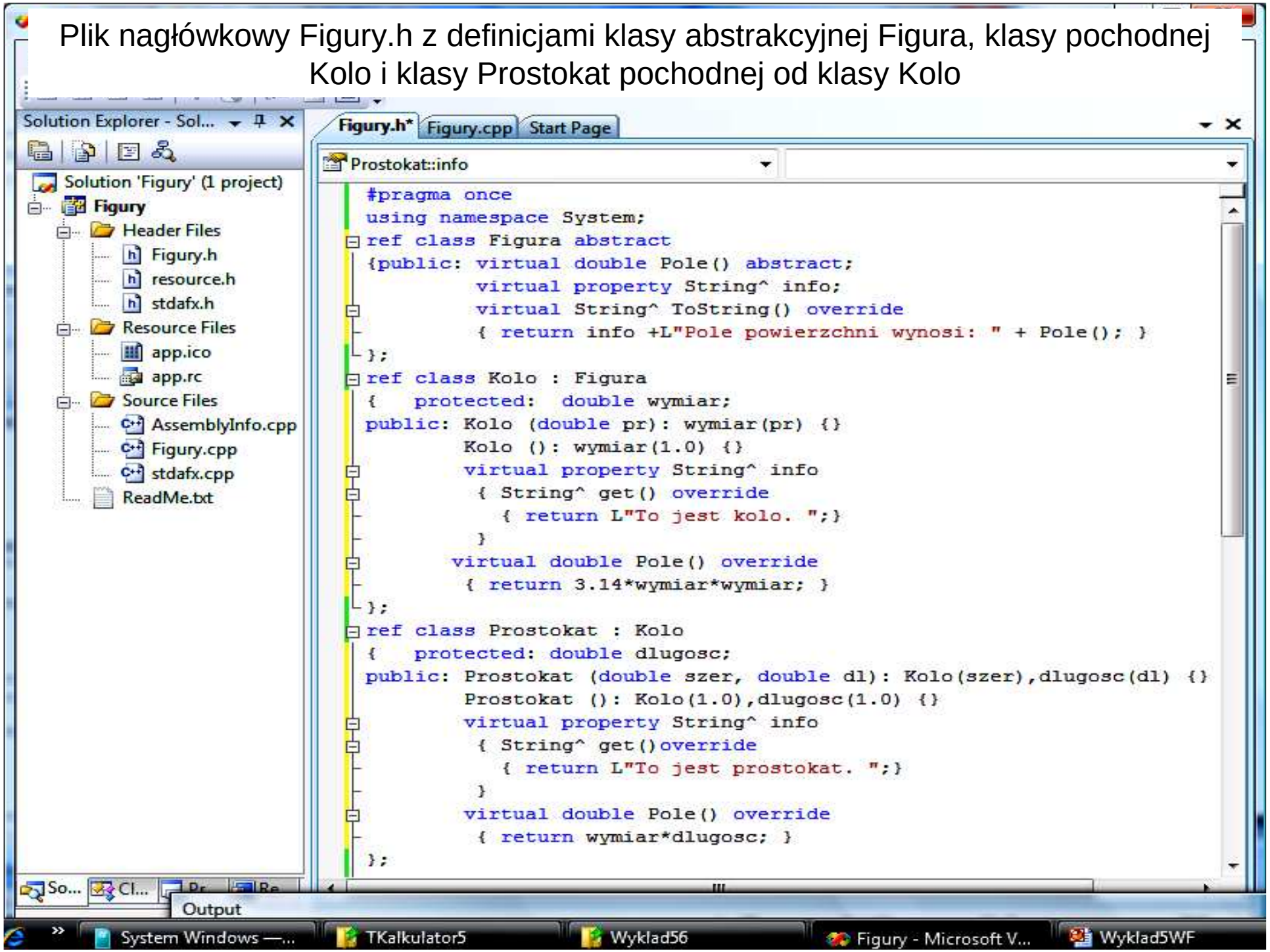
2.1. Przykład metod abstrakcyjnych i klasy abstrakcyjnej typu ref – projekt konsolowy Figury



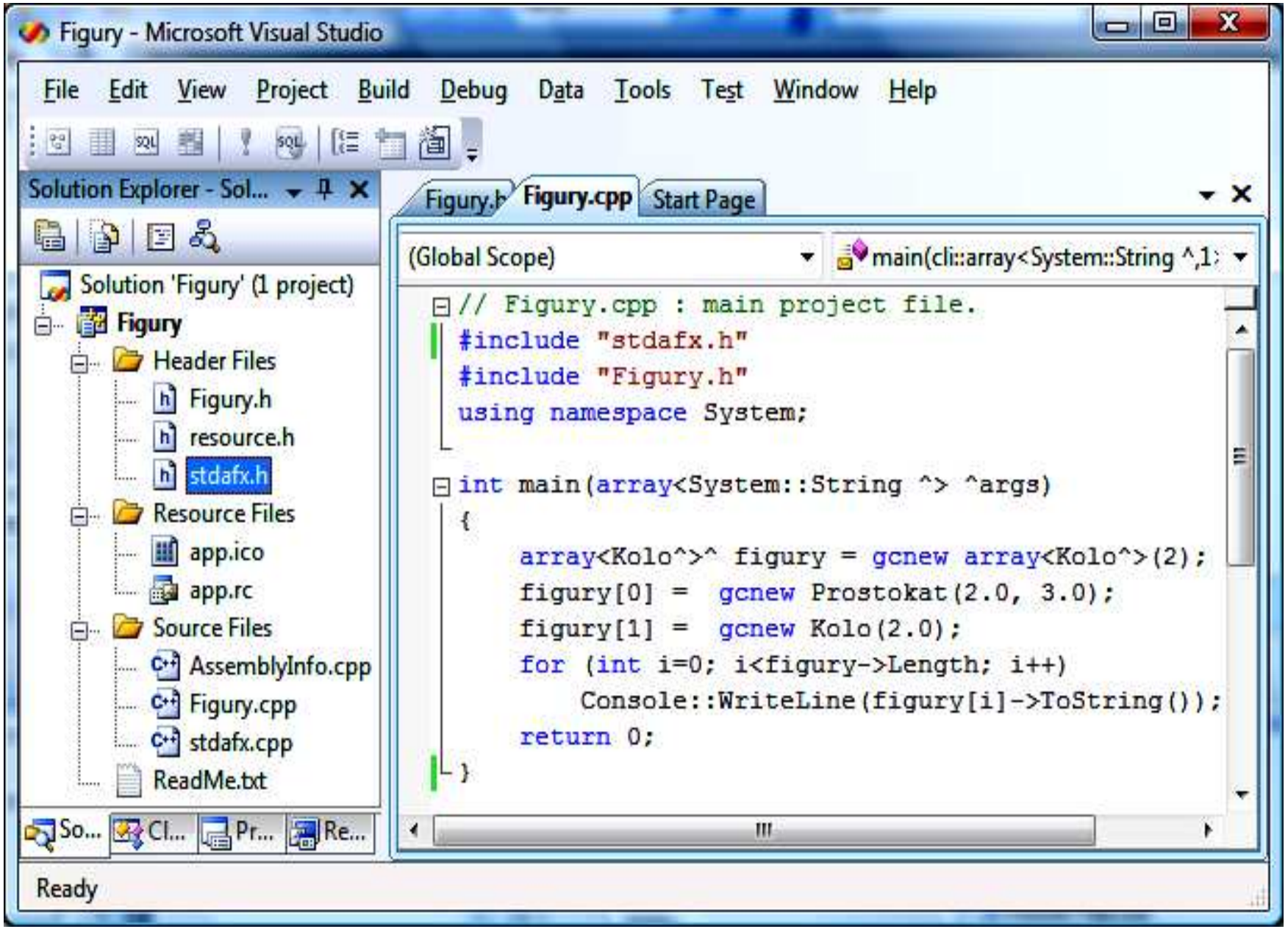
Dodanie pliku nagłówkowego Figury.h



Plik nagłówkowy Figury.h z definicjami klasy abstrakcyjnej Figura, klasy pochodnej Kolo i klasy Prostokat pochodnej od klasy Kolo



Program demonstrujący polimorfizm klasy abstrakcyjnej Figury



```
#pragma once
using namespace System;

ref class Figura abstract
{  public: virtual double Pole() abstract;
           virtual property String^ info;
           virtual String^ ToString() override
             {   return info +L"Pole powierzchni wynosi: " + Pole();  }
};

ref class Kolo : Figura
{ protected:  double wymiar;
  public:   Kolo (double pr) : wymiar(pr) { }
            Kolo () : wymiar(1.0) { }
            virtual property String^ info
              {
                  String^ get() override
                     { return L"To jest kolo. ";}
              }
            virtual double Pole() override
              {   return 3.14*wymiar*wymiar; }
};
```

Abstrakcyjna wirtualna metoda Pole

Abstrakcyjna klasa Figura

Wirtualna property info

Przedefiniowana wirtualna metoda ToString (po klasie System::Object)

Klasa Kolo dziedzicząca po klasie Figura

Konstruktory przeciążone

przedefiniowana wirtualna property info

przedefiniowana wirtualna metoda Pole

```
ref class Prostokat : Kolo
{   protected:
       double dlugosc;
public:
    Prostokat (double szer, double dl) : Kolo(szer), dlugosc(dl) { }
    Prostokat () : Kolo(1.0), dlugosc(1.0) { }

   virtual property String^ info
    {
          String^ get() override
           { return L"To jest prostokat. ";}
    }

  virtual double Pole() override
           { return wymiar*dlugosc; }
};
```

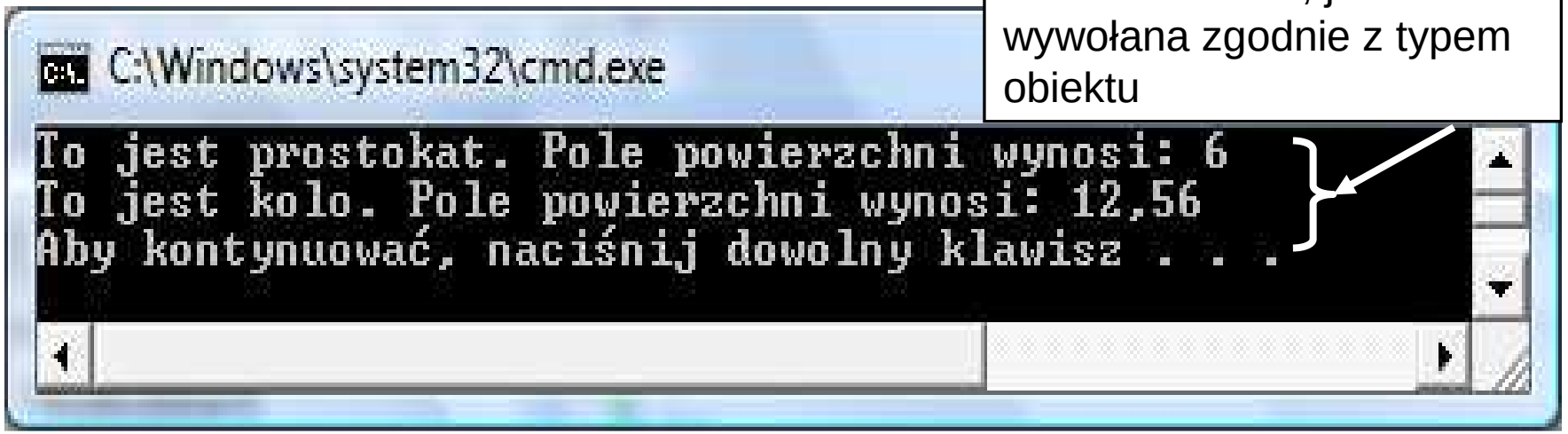
Klasa Prostokat dziedzicząca po klasie Kolo

Pole dziedziczone

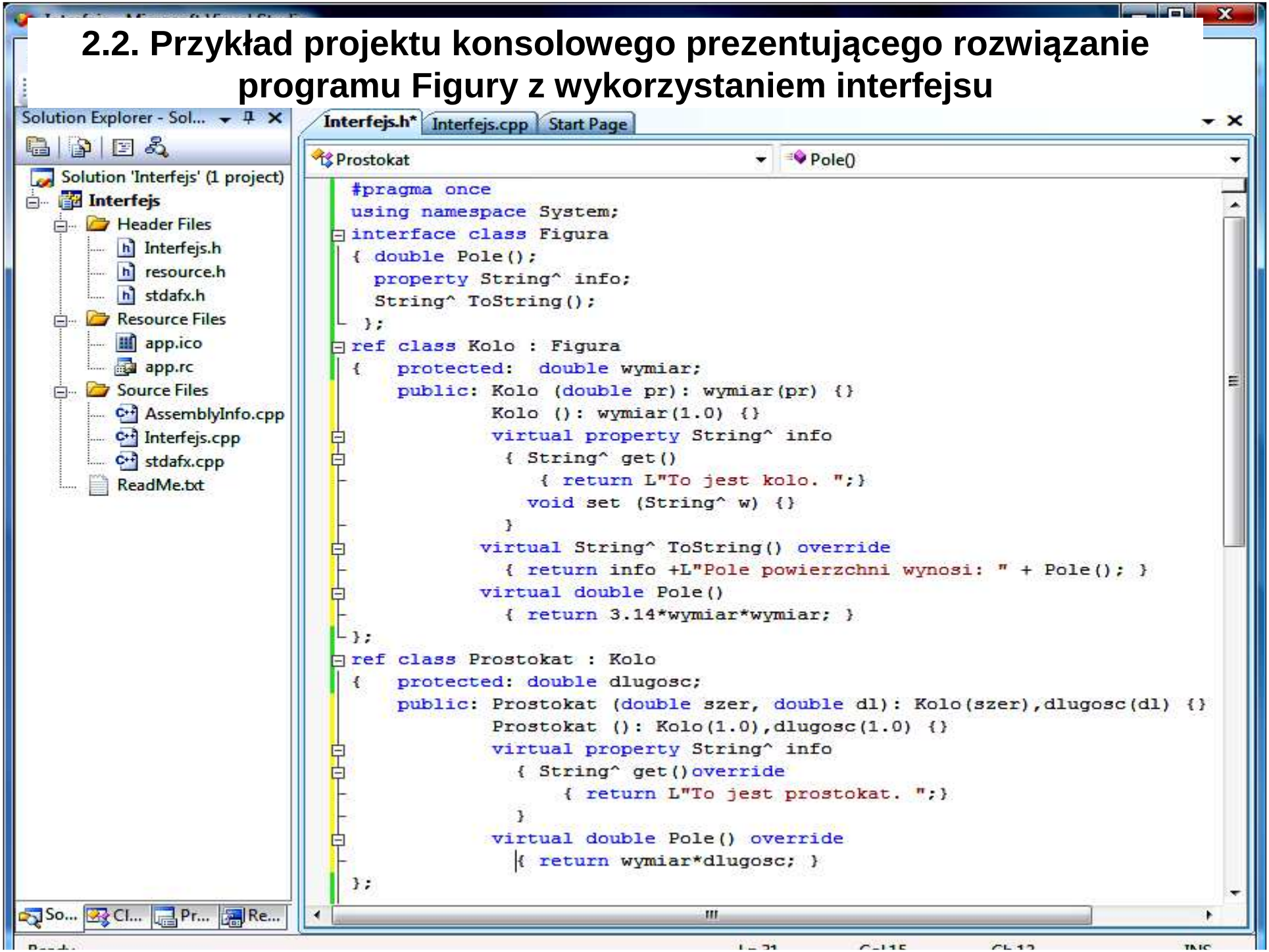
```
// Figury.cpp : main project file.

#include "stdafx.h"
#include „ Figury.h"
using namespace System;

int main(array<System::String ^> ^args)
{
    array<Kolo^>^ figury = gcnew array<Kolo^>(2);
    figury[0] =  gcnew Prostokat(2.0, 3.0);
    figury[1] =  gcnew Kolo(2.0);
    for (int i=0; i<figury->Length; i++)
            Console::WriteLine(figury[i]->ToString());
    return 0;
}
```

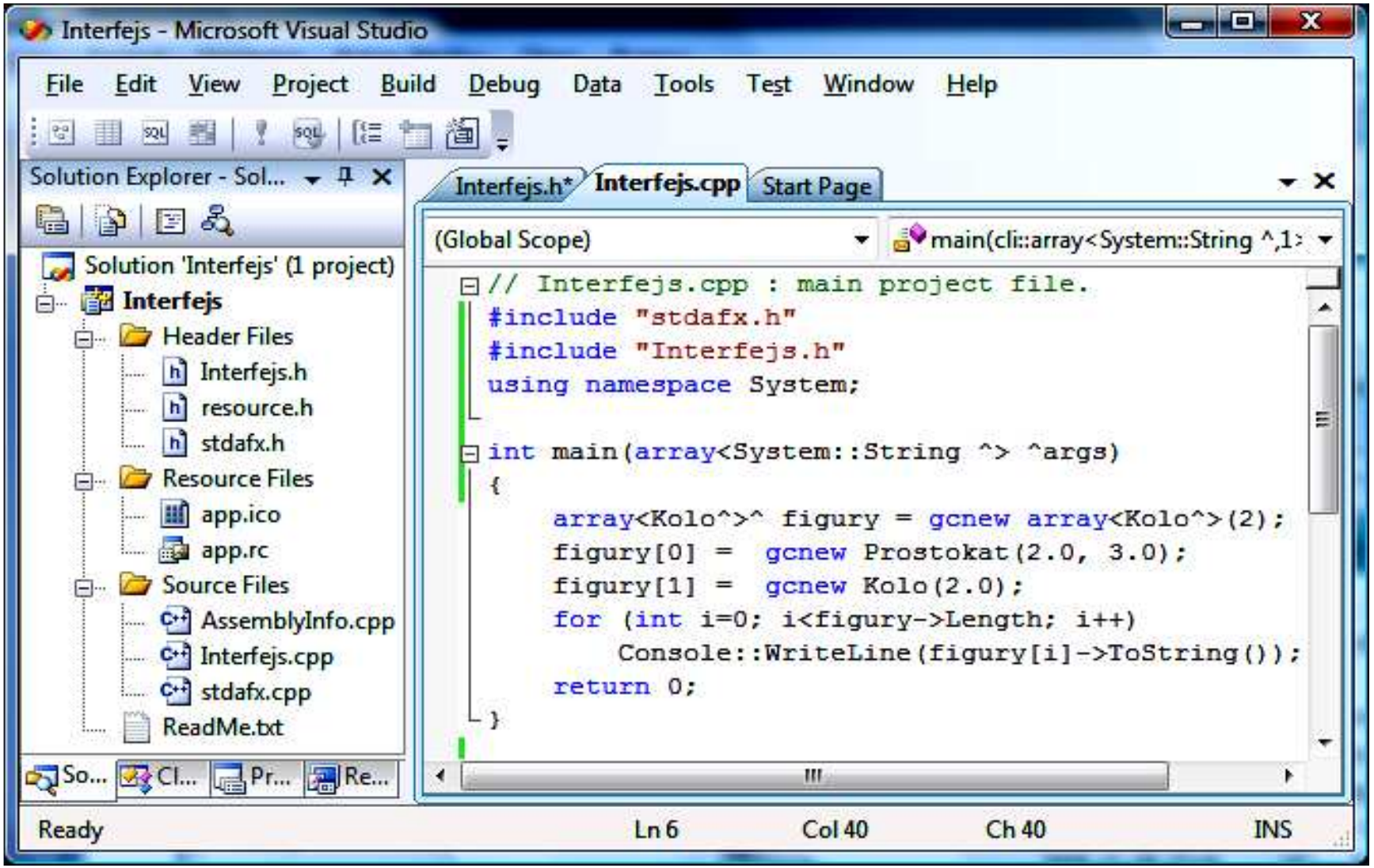
Tablica referencji do klasy Kolo może przechowywać referencje do obiektów klasy Kolo oraz Prostokat

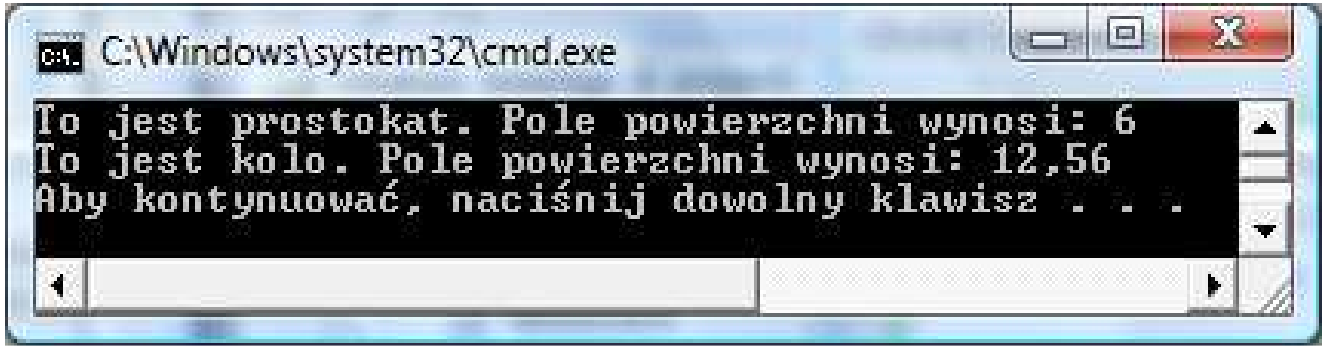
Wirtualna metoda ToString, przedefiniowana w klasach Kolo i Prostokat, jest wywołana zgodnie z typem obiektu



2.2. Przykład projektu konsolowego prezentującego rozwiązanie programu Figury z wykorzystaniem interfejsu




Interfejs - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Window Help
Solution Explorer - Sol...
Solution 'Interfejs' (1 project)
Interfejs
Header Files
Interfejs.h
resource.h
stdafx.h
Resource Files
app.ico
app.rc
Source Files
AssemblyInfo.cpp
Interfejs.cpp
stdafx.cpp
ReadMe.txt
So... Cl... Pr... Re...
Interfejs.h* Interfejs.cpp Start Page
(Global Scope)
main(cli::array<System::String ^,1>
// Interfejs.cpp : main project file.
#include "stdafx.h"
#include "Interfejs.h"
using namespace System;
int main(array<System::String ^> ^args)
{
array<Kolo^>^ figury = gcnew array<Kolo^>(2);
figury[0] = gcnew Prostokat(2.0, 3.0);
figury[1] = gcnew Kolo(2.0);
for (int i=0; i<figury->Length; i++)
Console::WriteLine(figury[i]->ToString());
return 0;
}
Ready
Ln 6
Col 40
Ch 40
INS


C:\Windows\system32\cmd.exe
To jest prostokat. Pole powierzchni wynosi: 6
To jest kolo. Pole powierzchni wynosi: 12,56
Aby kontynuować, naciśnij dowolny klawisz . . .

Rozwiązanie programu Figury za pomocą interfejsu Figura - kod klasy Prostokat i kod programu są takie same jak w programie z klasą abstrakcyjną (p. 2.1)

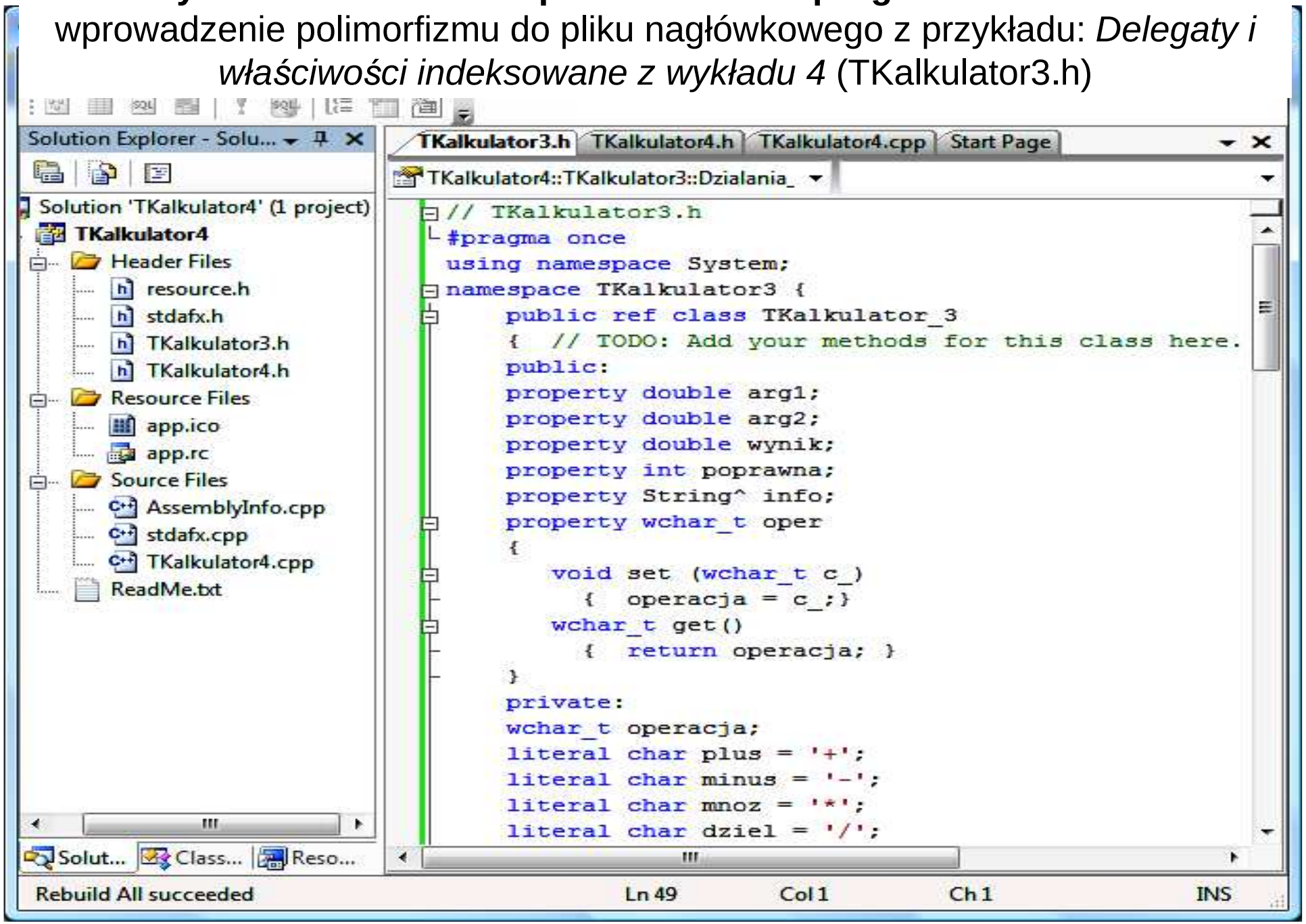
```
#pragma once
using namespace System;
interface class Figura
{    double Pole();
     property String^ info;
     String^ ToString();
 };

ref class Kolo : Figura
{   protected:  double wymiar;
    public:       Kolo (double pr) : wymiar(pr) { }
                  Kolo () : wymiar(1.0) { }
                  virtual property String^ info
                   { String^ get()
                        { return L"To jest kolo. ";}
                     void set (String^ w) { }
                   }
                 virtual String^ ToString() override
                     { return info +L"Pole powierzchni wynosi: " + Pole(); }
                 virtual double Pole()
                     { return 3.14*wymiar*wymiar; }
};
```

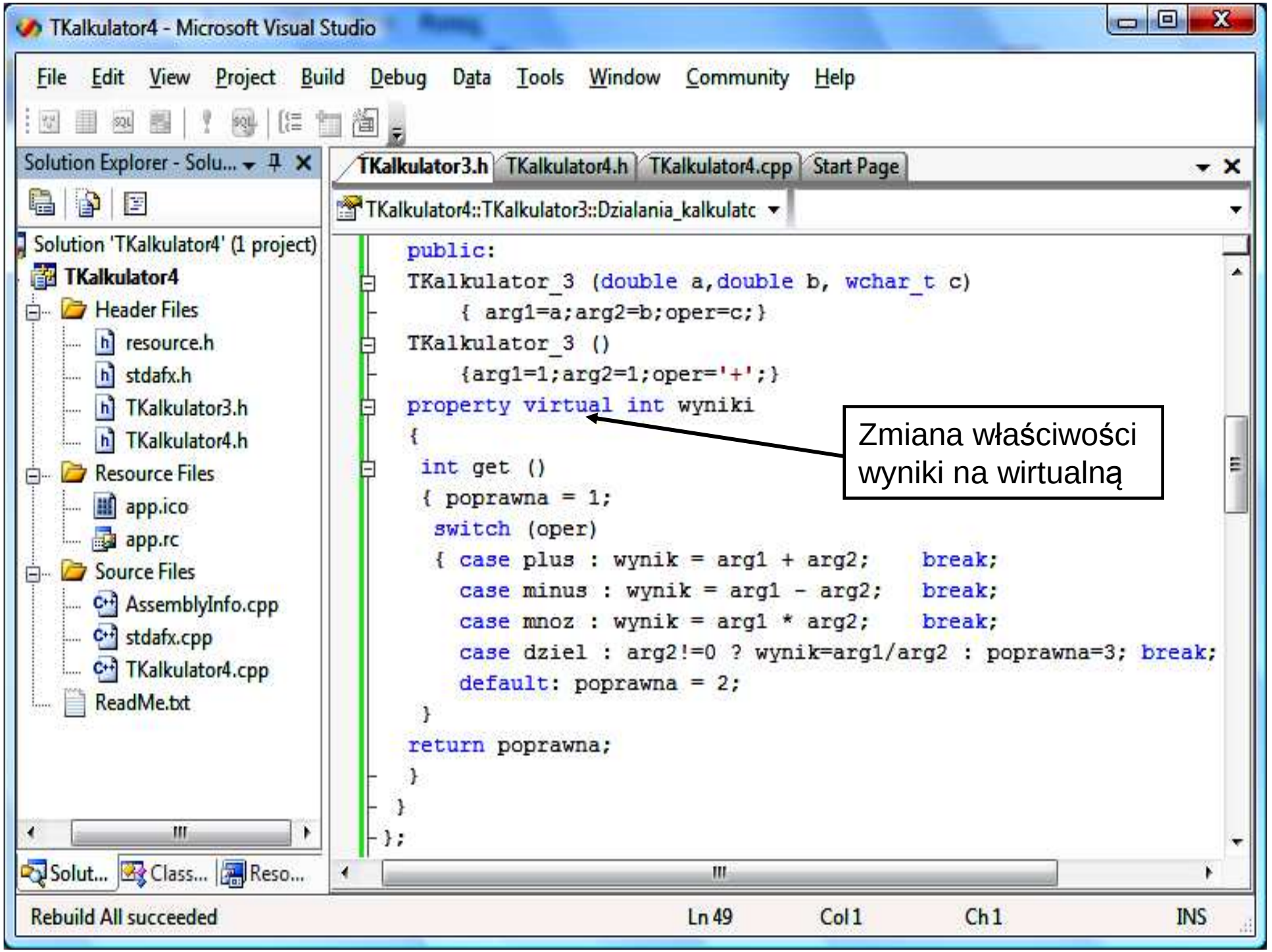
Deklaracje publicznych, abstrakcyjnych i wirtualnych metod w interfejsie

Należy implementować całą property info

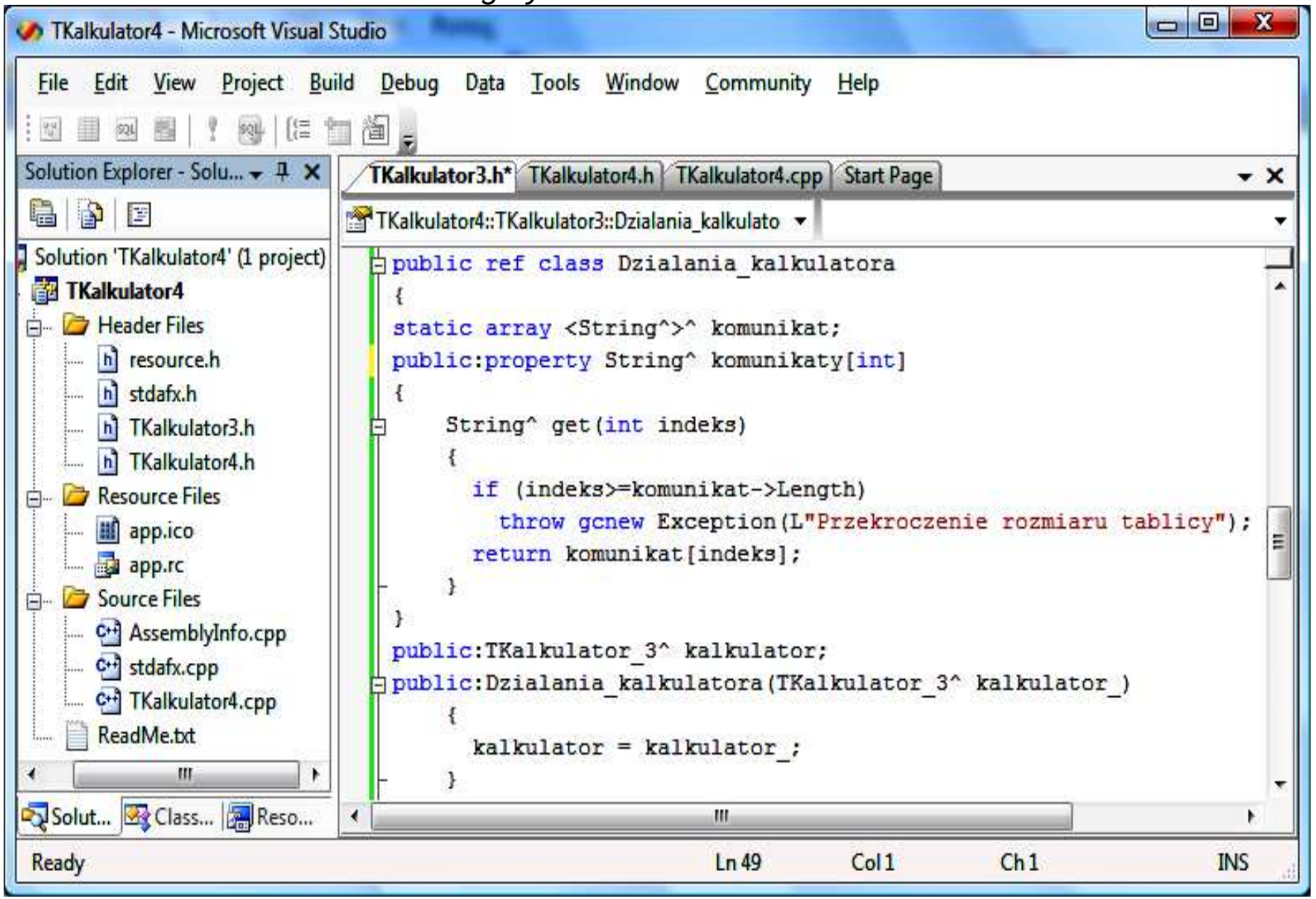
2.3. Przykład dziedziczenia i polimorfizmu w programie z kalkulatorem:

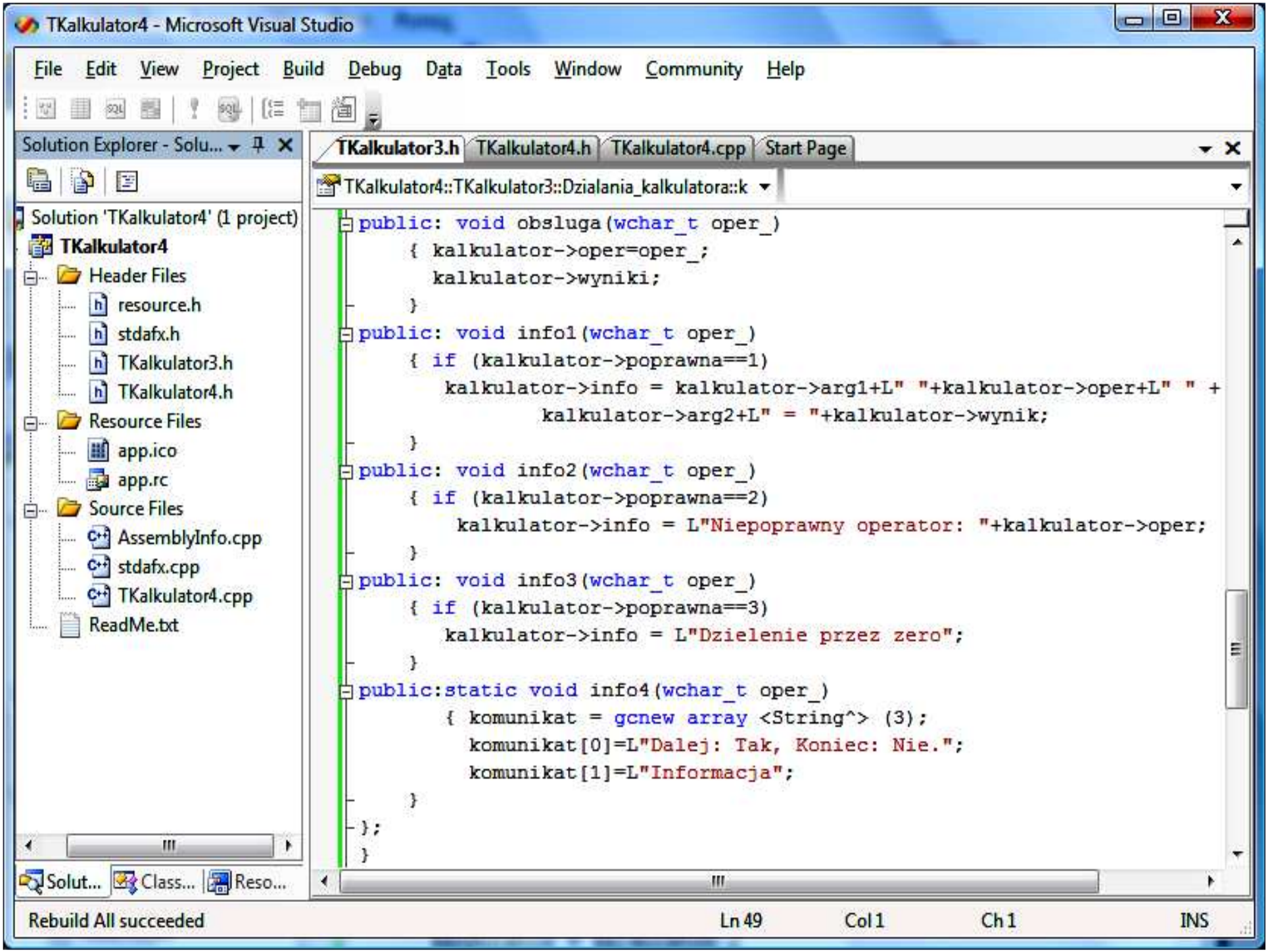
wprowadzenie polimorfizmu do pliku nagłówkowego z przykładu: *Delegaty i właściwości indeksowane z wykładu 4* (TKalkulator3.h)



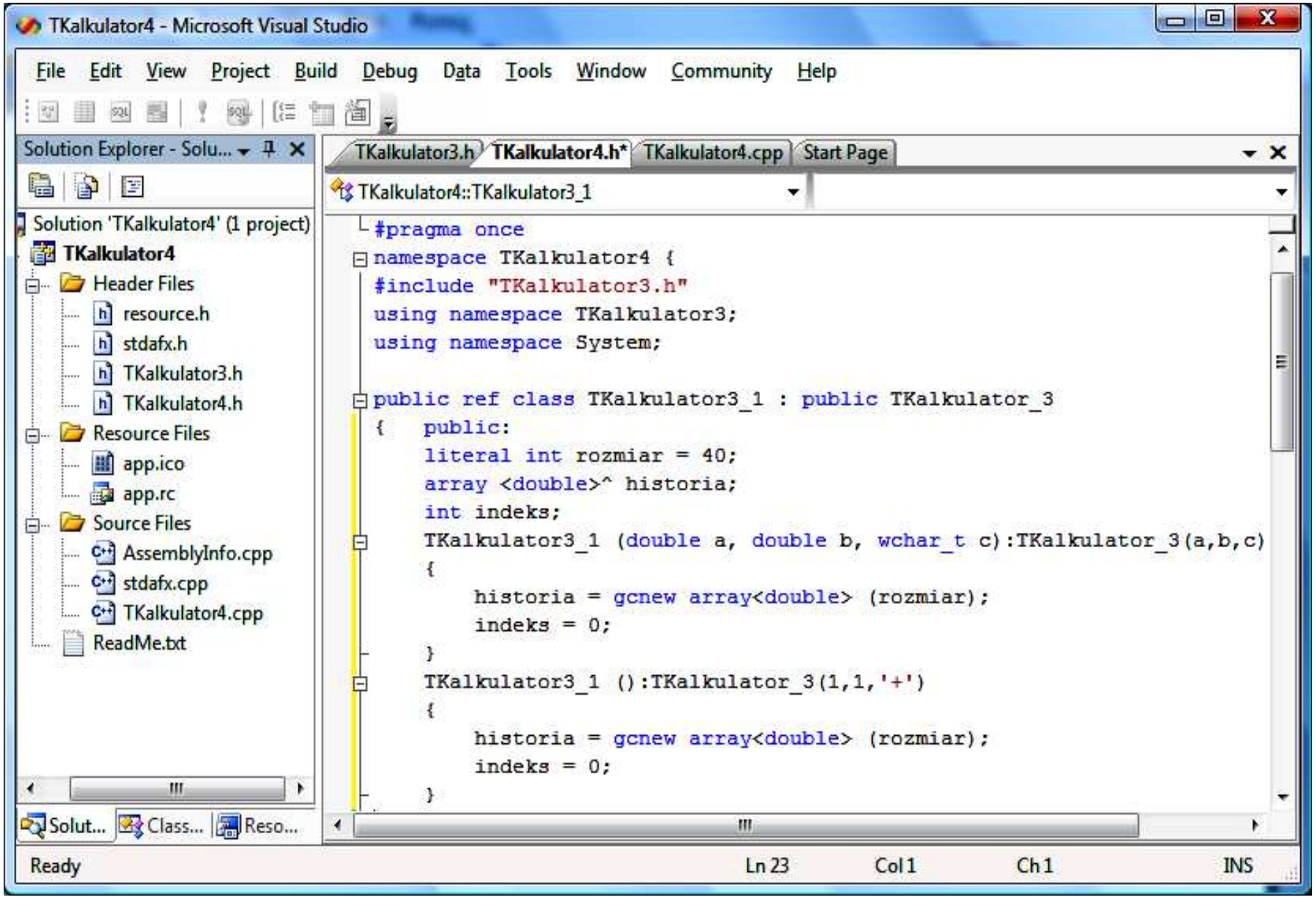

TKalkulator4 - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Solution Explorer - Solu...
Solution 'TKalkulator4' (1 project)
TKalkulator4
Header Files
resource.h
stdafx.h
TKalkulator3.h
TKalkulator4.h
Resource Files
app.ico
app.rc
Source Files
AssemblyInfo.cpp
stdafx.cpp
TKalkulator4.cpp
ReadMe.txt
Solut... Class... Reso...
TKalkulator3.h TKalkulator4.h TKalkulator4.cpp Start Page
TKalkulator4::TKalkulator3::Dzialania_kalkulatc
public:
TKalkulator_3 (double a,double b, wchar_t c)
{ arg1=a;arg2=b;oper=c;}
TKalkulator_3 ()
{arg1=1;arg2=1;oper='+';}
property virtual int wyniki
{
int get ()
{ poprawna = 1;
switch (oper)
{ case plus : wynik = arg1 + arg2; break;
case minus : wynik = arg1 - arg2; break;
case mnoz : wynik = arg1 * arg2; break;
case dziel : arg2!=0 ? wynik=arg1/arg2 : poprawna=3; break;
default: poprawna = 2;
}
return poprawna;
}
}
};
Zmiana właściwości wyniki na wirtualną
Rebuild All succeeded
Ln 49
Col 1
Ch 1
INS

Definicja klasy ***Działania_kalkulatora*** identyczna jak w przykładzie z wykładu 4: *Delegaty i właściwości indeksowane*

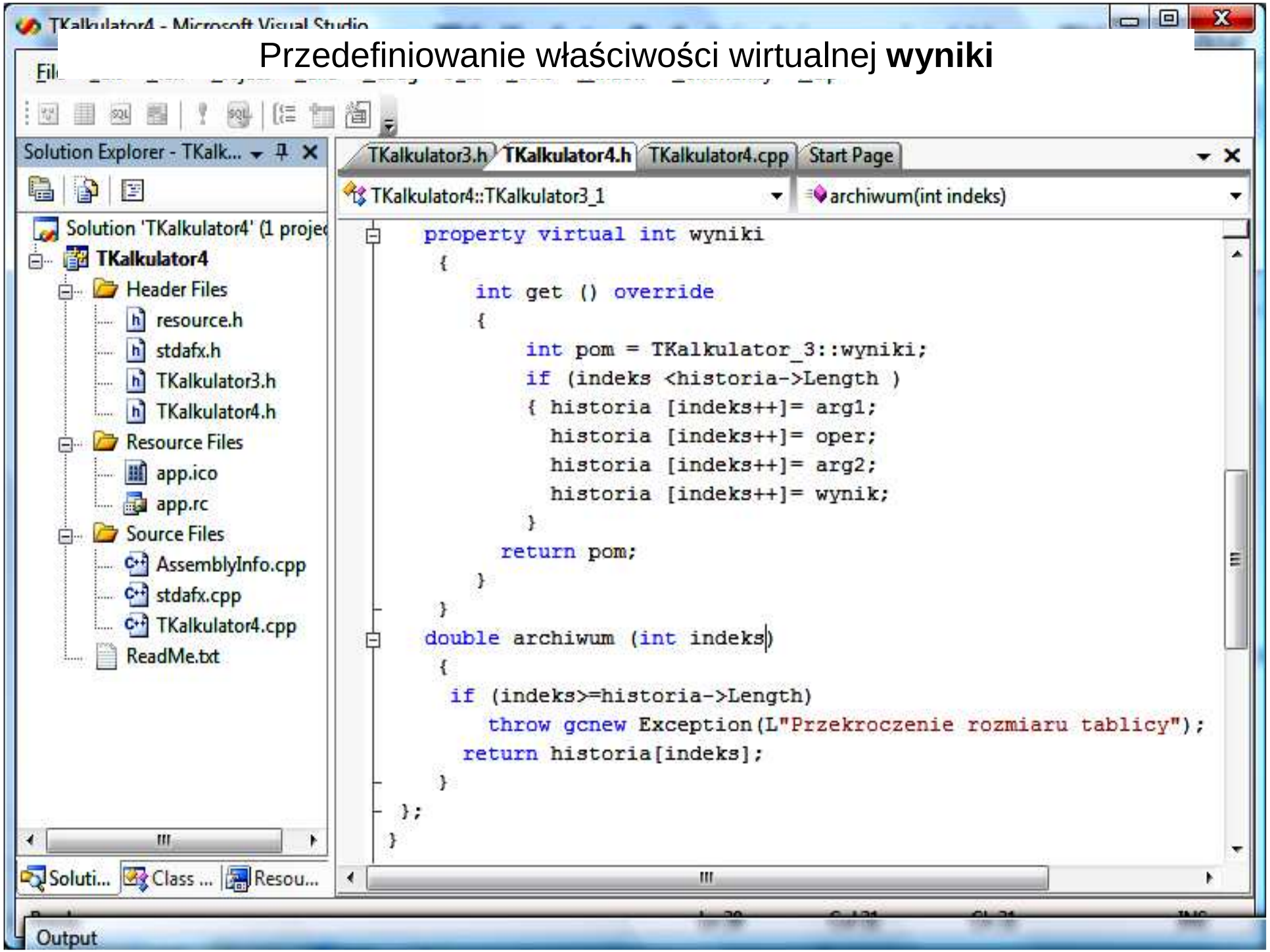



TKalkulator4 - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Solution Explorer - Solu...
Solution 'TKalkulator4' (1 project)
TKalkulator4
Header Files
resource.h
stdafx.h
TKalkulator3.h
TKalkulator4.h
Resource Files
app.ico
app.rc
Source Files
AssemblyInfo.cpp
stdafx.cpp
TKalkulator4.cpp
ReadMe.txt
Solut... Class... Reso...
TKalkulator3.h TKalkulator4.h TKalkulator4.cpp Start Page
TKalkulator4::TKalkulator3::Dzialania_kalkulatora::k
public: void obsluga(wchar_t oper_)
{ kalkulator->oper=oper_;
kalkulator->wyniki;
}
public: void info1(wchar_t oper_)
{ if (kalkulator->poprawna==1)
kalkulator->info = kalkulator->arg1+L" "+kalkulator->oper+L" " +
kalkulator->arg2+L" = "+kalkulator->wynik;
}
public: void info2(wchar_t oper_)
{ if (kalkulator->poprawna==2)
kalkulator->info = L"Niepoprawny operator: "+kalkulator->oper;
}
public: void info3(wchar_t oper_)
{ if (kalkulator->poprawna==3)
kalkulator->info = L"Dzielenie przez zero";
}
public:static void info4(wchar_t oper_)
{ komunikat = gcnew array <String^> (3);
komunikat[0]=L"Dalej: Tak, Koniec: Nie.";
komunikat[1]=L"Informacja";
}
};
}
Rebuild All succeeded
Ln 49
Col 1
Ch 1
INS

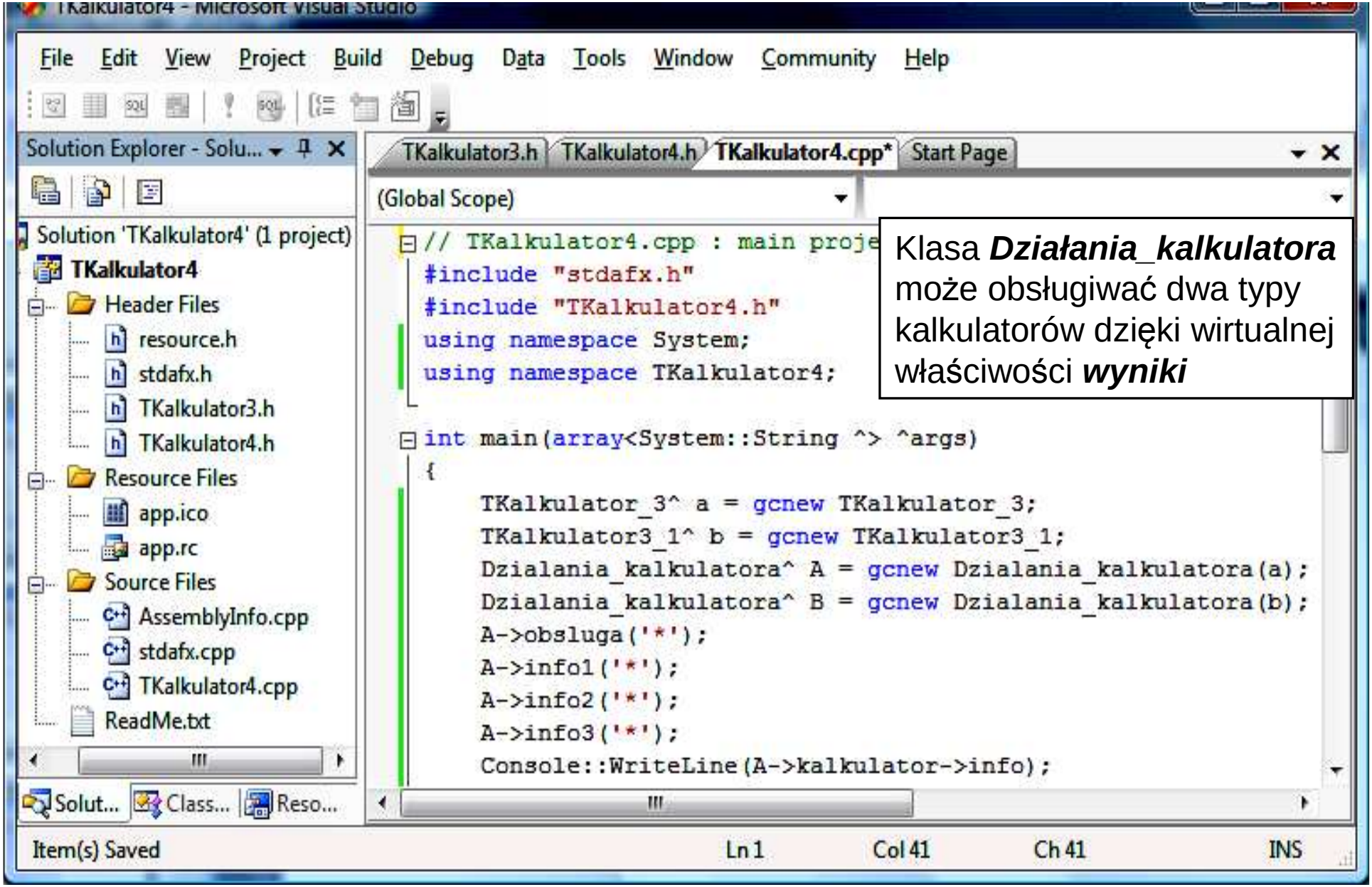
Definicja klasy **TKalkulator3_1** dziedziczącej od klasy **TKalukator_3**

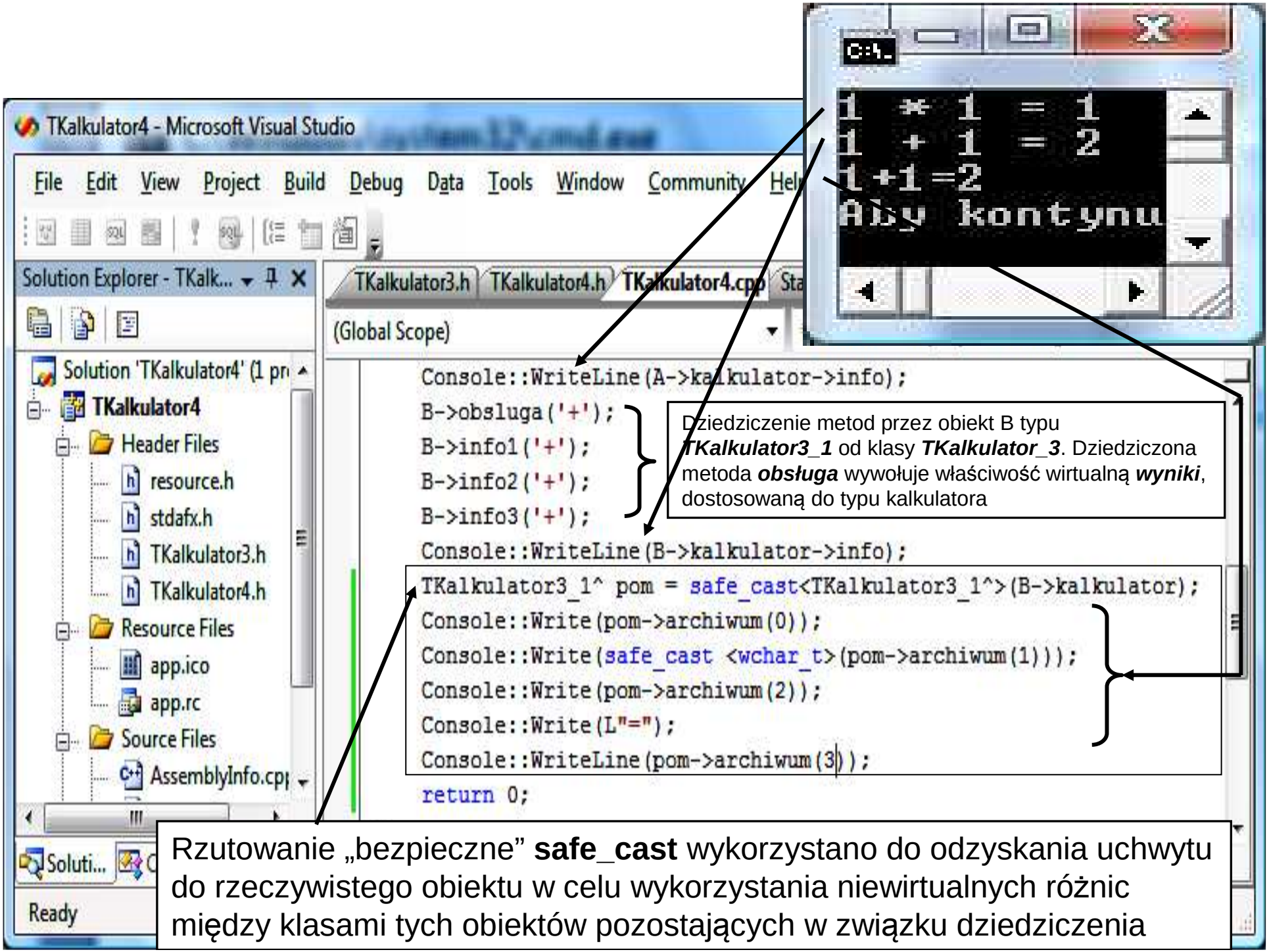


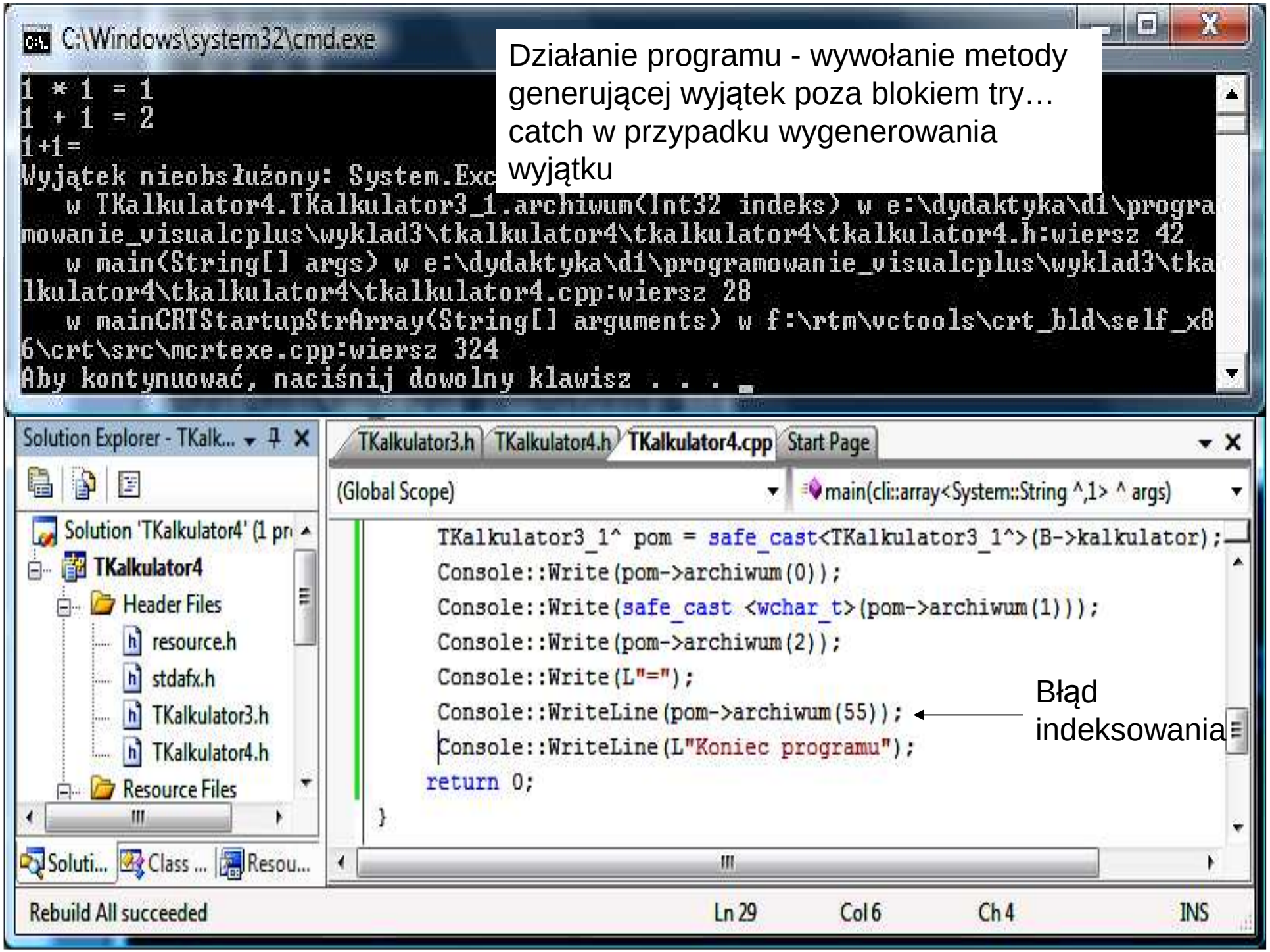
Przedefiniowanie właściwości wirtualnej **wyniki**

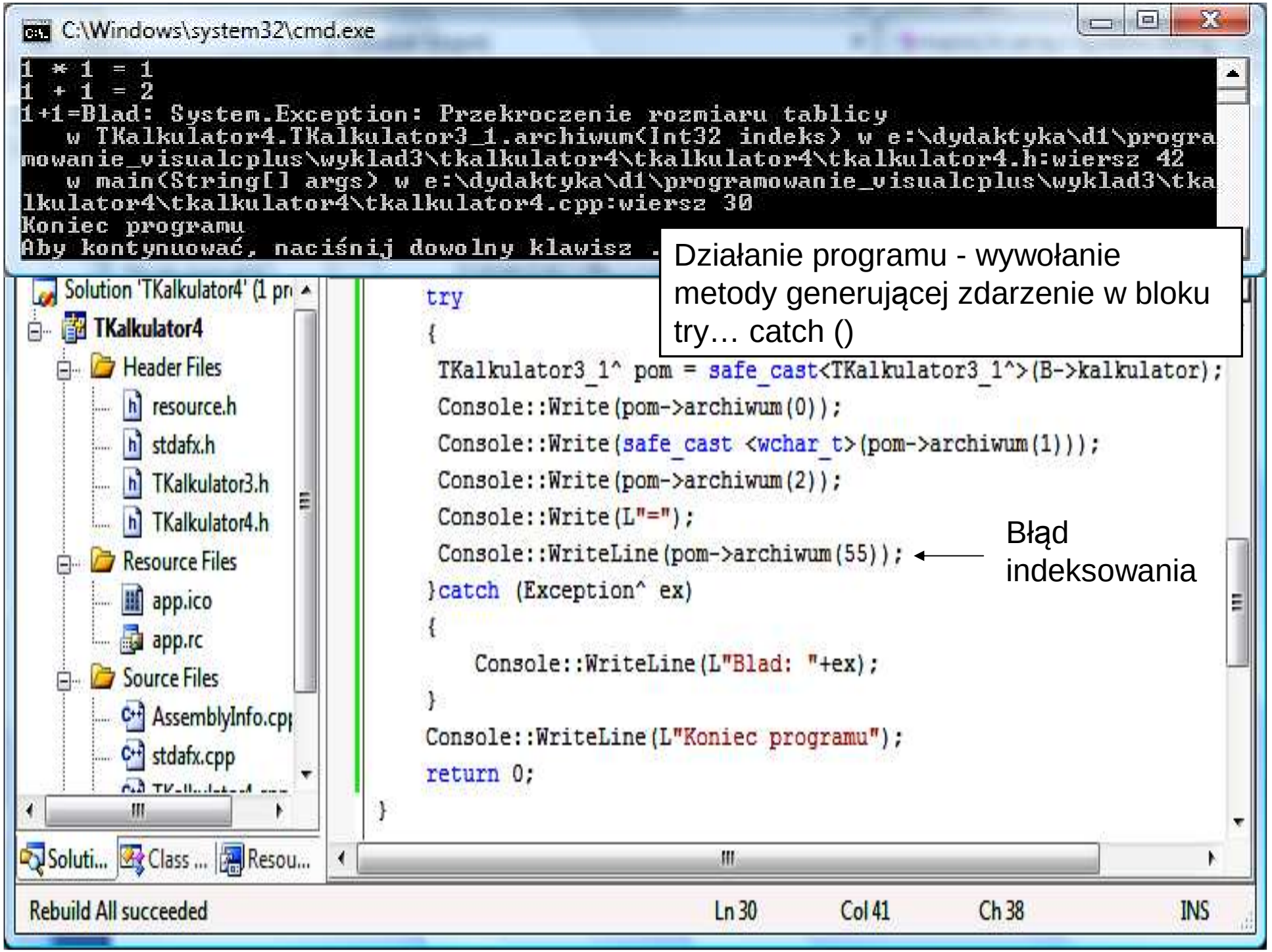


Program prezentujący działanie dziedziczenia i polimorfizmu obiektów typu ***TKalkulator_3*** i ***TKalkulator3_1*** w obiektach klasy ***Dzialania_kalkulatora***




TKalkulator4 - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Solution Explorer - TKalk...
Solution 'TKalkulator4' (1 pr
TKalkulator4
Header Files
resource.h
stdafx.h
TKalkulator3.h
TKalkulator4.h
Resource Files
app.ico
app.rc
Source Files
AssemblyInfo.cp
Soluti...
Ready
TKalkulator3.h
TKalkulator4.h
TKalkulator4.cpp
(Global Scope)
Console::WriteLine(A->kalkulator->info);
B->obsluga('+');
B->info1('+');
B->info2('+');
B->info3('+');
Console::WriteLine(B->kalkulator->info);
TKalkulator3_1^ pom = safe_cast<TKalkulator3_1^>(B->kalkulator);
Console::Write(pom->archiwum(0));
Console::Write(safe_cast <wchar_t>(pom->archiwum(1)));
Console::Write(pom->archiwum(2));
Console::Write(L"=");
Console::WriteLine(pom->archiwum(3));
return 0;
C:\
1 * 1 = 1
1 + 1 = 2
1+1=2
Aby kontynu
Dziedziczenie metod przez obiekt B typu TKalkulator3_1 od klasy TKalkulator_3. Dziedziczona metoda obsługa wywołuje właściwość wirtualną wyniki, dostosowaną do typu kalkulatora
Rzutowanie „bezpieczne” safe_cast wykorzystano do odzyskania uchwytu do rzeczywistego obiektu w celu wykorzystania niewirtualnych różnic między klasami tych obiektów pozostających w związku dziedziczenia


C:\Windows\system32\cmd.exe
1 * 1 = 1
1 + 1 = 2
1+1=
Wyjątek nieobsłużony: System.Exc
w TKalkulator4.TKalkulator3_1.archiwum(Int32 indeks) w e:\dydaktyka\d1\progra
mowanie_visualcplus\wyklad3\tkalkulator4\tkalkulator4\tkalkulator4.h:wiersz 42
w main(String[] args) w e:\dydaktyka\d1\programowanie_visualcplus\wyklad3\tka
lkulator4\tkalkulator4\tkalkulator4.cpp:wiersz 28
w mainCRTStartupStrArray(String[] arguments) w f:\rtm\vctools\crt_bld\self_x8
6\crt\src\mcrtexe.cpp:wiersz 324
Aby kontynuować, naciśnij dowolny klawisz . . .
Działanie programu - wywołanie metody generującej wyjątek poza blokiem try… catch w przypadku wygenerowania wyjątku
Solution Explorer - TKalk...
Solution 'TKalkulator4' (1 pr
TKalkulator4
Header Files
resource.h
stdafx.h
TKalkulator3.h
TKalkulator4.h
Resource Files
Soluti...
Class ...
Resou...
TKalkulator3.h
TKalkulator4.h
TKalkulator4.cpp
Start Page
(Global Scope)
main(cli::array<System::String ^,1> ^ args)
TKalkulator3_1^ pom = safe_cast<TKalkulator3_1^>(B->kalkulator);
Console::Write(pom->archiwum(0));
Console::Write(safe_cast <wchar_t>(pom->archiwum(1)));
Console::Write(pom->archiwum(2));
Console::Write(L"=");
Console::WriteLine(pom->archiwum(55));
Console::WriteLine(L"Koniec programu");
return 0;
}
Błąd indeksowania
Rebuild All succeeded
Ln 29
Col 6
Ch 4
INS


C:\Windows\system32\cmd.exe
1 * 1 = 1
1 + 1 = 2
1+1=Blad: System.Exception: Przekroczenie rozmiaru tablicy
w TKalkulator4.TKalkulator3_1.archiwum(Int32 indeks) w e:\dydaktyka\d1\programowanie_visualcplus\wyklad3\tkalkulator4\tkalkulator4\tkalkulator4.h:wiersz 42
w main(String[] args) w e:\dydaktyka\d1\programowanie_visualcplus\wyklad3\tkalkulator4\tkalkulator4\tkalkulator4.cpp:wiersz 30
Koniec programu
Aby kontynuować, naciśnij dowolny klawisz .
Działanie programu - wywołanie metody generującej zdarzenie w bloku try… catch ()
Solution 'TKalkulator4' (1 pr
TKalkulator4
Header Files
resource.h
stdafx.h
TKalkulator3.h
TKalkulator4.h
Resource Files
app.ico
app.rc
Source Files
AssemblyInfo.cp
stdafx.cpp
try
{
TKalkulator3_1^ pom = safe_cast<TKalkulator3_1^>(B->kalkulator);
Console::Write(pom->archiwum(0));
Console::Write(safe_cast <wchar_t>(pom->archiwum(1)));
Console::Write(pom->archiwum(2));
Console::Write(L"=");
Console::WriteLine(pom->archiwum(55));
Błąd indeksowania
}catch (Exception^ ex)
{
Console::WriteLine(L"Blad: "+ex);
}
Console::WriteLine(L"Koniec programu");
return 0;
}
Soluti...
Class ...
Resou...
Rebuild All succeeded
Ln 30
Col 41
Ch 38
INS