

Języki i paradygmaty programowania

doc. dr inż. Tadeusz Jeleniewski

e-mail:

[*t.jeleniewski@neostrada.pl*](mailto:t.jeleniewski@neostrada.pl)

[*tadeusz.jeleniewski@pwr.wroc.pl*](mailto:tadeusz.jeleniewski@pwr.wroc.pl)

[*http://www.tjeleniewski.wstt.edu.pl*](http://www.tjeleniewski.wstt.edu.pl)

Treści kształcenia:

Paradygmaty programowania. Wybrane zagadnienia programowania obiektowego.

Efekty kształcenia - umiejętności i kompetencje:

oceny przydatności wybranych paradygmatów i związanych z nimi środowisk programistycznych do rozwiązywania różnego typu problemów:

projektowania, implementacji, testowania na przykładzie wybranego języka i platformy (C# i .NET)

Literatura

- Perry S.C. - C# i .NET. Wydawnictwo Helion, 2006
- Liberty J. – Programowanie C#. Wyd. Helion, 2006
- Boduch A. – Wstęp do programowania w języku C#. Wyd. Helion, 2006
- Troelsen A. – Język C# i platforma .NET. Wyd. Mikom, 2006
- Hoffman K. – Visual C# 2005, księga eksperta. Wyd. Helion, 2007
- Liberty J., Hurwitz D., Programowanie ASP.NET. Wyd. HELION, 2007

1	Przegląd i charakterystyka narzędzi RAD do tworzenia oprogramowania użytkowego.
2	Ogólna charakterystyka środowiska Microsoft Visual Studio .NET i języka C#
3	Różnice pomiędzy językiem C++ a C#
4	Struktura programu w języku C#. Program jako obiekt klasy Application
5	Tworzenie programów konsolowych z wykorzystaniem środowiska MS Visual Studio 2008
6	Tworzenie aplikacji windows'owych z wykorzystaniem komponentów biblioteki Windows Forms
7	Platforma ADO.NET – programowanie aplikacji "bazodanowych"
8	Technologia ASP.NET i jej zastosowanie do tworzenia serwisów internetowych

Formy dydaktyczne:

-Wykład - termin: poniedziałki godz. 8:00 – 9:35, sala 14

-Laboratorium – poniedziałki parzyste, godz. 11:30 – 13:05, sala 16

Zaliczenie przedmiotu na stopień – na podstawie kolokwium na przedostatnim wykładzie w semestrze.

Ostateczna ocena będzie średnią arytmetyczną oceny z kolokwium i oceny z laboratorium pod warunkiem, że obydwie te oceny będą pozytywne.

Wykład 1

- **Narzędzia RAD do budowy aplikacji**
- **Elementy składowe aplikacji**
- **Charakterystyka języka C#**
- **Język C# i środowisko IDE**

Spośród znajdujących się aktualnie na rynku narzędzi wspomagających tworzenie programów użytkowych godne uwagi są systemy oprogramowania:

- Borland Delphi,
- Borland C++ Builder,
- Sybase Power++,
- Microsoft Visual C++,
- Microsoft Visual .NET C#

Mówimy o nich, że są to **środowiska programistyczne**.

Wszystkie te systemy działają w 32 – bitowym środowisku Windows 95/98/2000/XP NT i wykorzystują jego mechanizmy GUI (*Graphic User Interface*). Narzędzia tej klasy określa się wspólnym mianem **RAD** (*Rapid Application Development* – błyskawiczne tworzenie aplikacji).

System Delphi wykorzystuje obiektową wersję języka Pascal (tzw. Object Pascal). Do kodowania programów (automatycznego lub „ręcznego”) w systemie C++ Builder wykorzystano wersję „borlandowską” języka C++, a system Sybase Power++ oparto na wersji Watcom 11 języka C++.

Elementy składowe aplikacji

Podstawowe elementy, z których budowane są aplikacje to:

- *formularze,*
- *obiekty (komponenty),*
- *zdarzenia.*

W systemach C++ Builder, Delphi, Power++, C# każdy formularz jest zdefiniowany jako *klasa*.

Fakt ten ma następujące konsekwencje:

1. Formularz jest typem danych, a nie obiektem danych. Jeden z formularzy składających się na aplikację jest *formularzem głównym*. Jest on tworzony automatycznie w momencie uruchomienia programu i służy jako *okienko inicjacyjne*.
2. Można utworzyć wiele obiektów tego samego typu *formularza*.

3. Każda klasa formularza ma związane ze sobą *własności*. Jedne z nich dotyczą postaci formularza widocznego na ekranie (kolor, wielkość itp.). inne z nich mają wpływ na zachowanie się – np. określają, czy wielkość formularza może być zmieniana. Projektując formularz określamy wartości początkowe jego własności. Zestaw wartości początkowych jest używany zawsze wtedy, gdy nasz program tworzy nowy obiekt klasy *formularz*. W czasie wykonywania programu wiele z tych własności można zmienić.

- 4. Każda klasa formularza ma zbiór związanych z nią *metod*. *Metoda* jest funkcją, dzięki której można wykonać akcję z użyciem formularza. Przykładowo, formularz ma metody sprawdzania zmian własności formularza.**

- 5. We wszystkich tych systemach istnieje możliwość dodania własnej metody do klasy formularza. Jest to przydatne wtedy, gdy chcemy zdefiniować procedurę, która może być użyta przez inną funkcję wewnątrz klasy lub gdy chcemy zapewnić kontrolowany dostęp do klasy z obiektów znajdujących się poza klasą.**

Język C# i środowisko IDE

Literatura: Liberty J., Programowanie C#. Wyd. Helion, Gliwice, 2006

Platforma .NET

– zestaw języków programowania:

C++, C#, Visual Basic, Visual J#, Jscript.NET i zestaw narzędzi programistycznych Visual Studio .NET, biblioteka klas i środowisko uruchomieniowe CLR (*Common Language Routine*) do wykonywania programów przeznaczonych na tę platformę

- dwie generacje serwerów .NET Enterprise
- nowe urządzenia przystosowane do platformy .NET (telefony komórkowe, konsole do gier itp.).

Wykonywanie programów na platformie .NET możliwe jest po zainstalowaniu w systemie mechanizmu .NET Framework.

Jego najważniejszym elementem jest środowisko uruchomieniowe CLR, które składa się z tzw. „maszyny wirtualnej” (podobnej do VM Javy). CLR tworzy obiekty, przydziela im pamięć, sprawdza bezpieczeństwo, wykonuje polecenia i odzyskuje pamięć.

Jego częścią jest wspólny dla języków programowania system typów.

Kompilator programów na platformę NET **nie tworzy plików wykonywalnych** – przekłada źródła na ciągi instrukcji standardowego języka pośredniego MSIL (*Microsoft Intermediate Language*).

Dopiero w środowisku CLR następuje ich przekształcenie na kod maszynowy i wykonanie.

Kompilacja kodu źródłowego na kod pośredni następuje w czasie budowy projektu. Kod pośredni zostaje zapisany na dysku w pliku .exe (nie należy go mylić z plikiem wykonywalnym w innych systemach programowania).

W momencie uruchomienia programu następuje ponowna kompilacja programu – tzw. "kompilacja w locie" (JIT - *Just In Time*).

W wyniku kompilacji JIT powstaje kod maszynowy, który wykonuje procesor maszyny.

Platforma .NET

Usługi Web Services

Interfejs aplikacji
internetowych

Interfejs
aplikacji Windows

Klasy do obsługi danych i formatu XML
(ADO.NET, SQL, XSLT, XPath, XML itd.)

Klasy bazowe
(obsługa wejścia i wyjścia, łańcuchów znaków, sieci, bezpieczeństwa, wątków, tekstu, odzwierciedlania, kolekcji itd.)

Wspólne środowisko uruchomieniowe
(usuwanie błędów, obsługa wyjątków, sprawdzanie typów, kompilacja JIT)

Platforma Windows

Charakterystyka języka C#

- prosty bo zawiera tylko około 80 słów kluczowych i kilkanaście wbudowanych typów,
- zestaw operatorów i instrukcji podobny do C++, reguły składni, leksykalne itp. również zbliżone do C++,
- programy w C# budowane są z obiektów klas z bogatej biblioteki oraz z obiektów klas „własnych” programisty,
- wszystkie elementy klasy znajdują się w jej deklaracji, niepotrzebne są pliki nagłówkowe ani pliki definicji interfejsu,
- możliwe jest automatyczne tworzenie dokumentacji projektu z wykorzystaniem XML

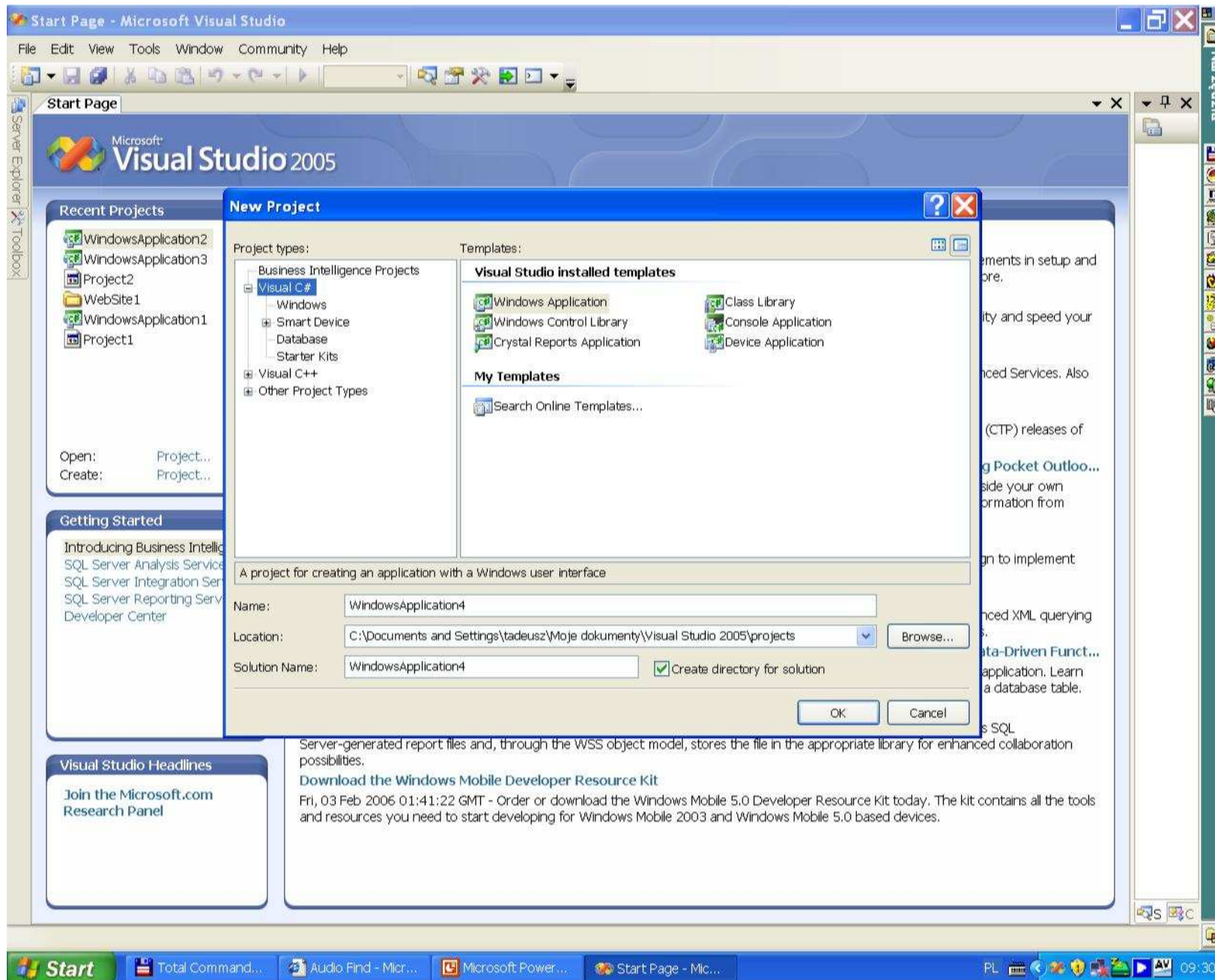
```

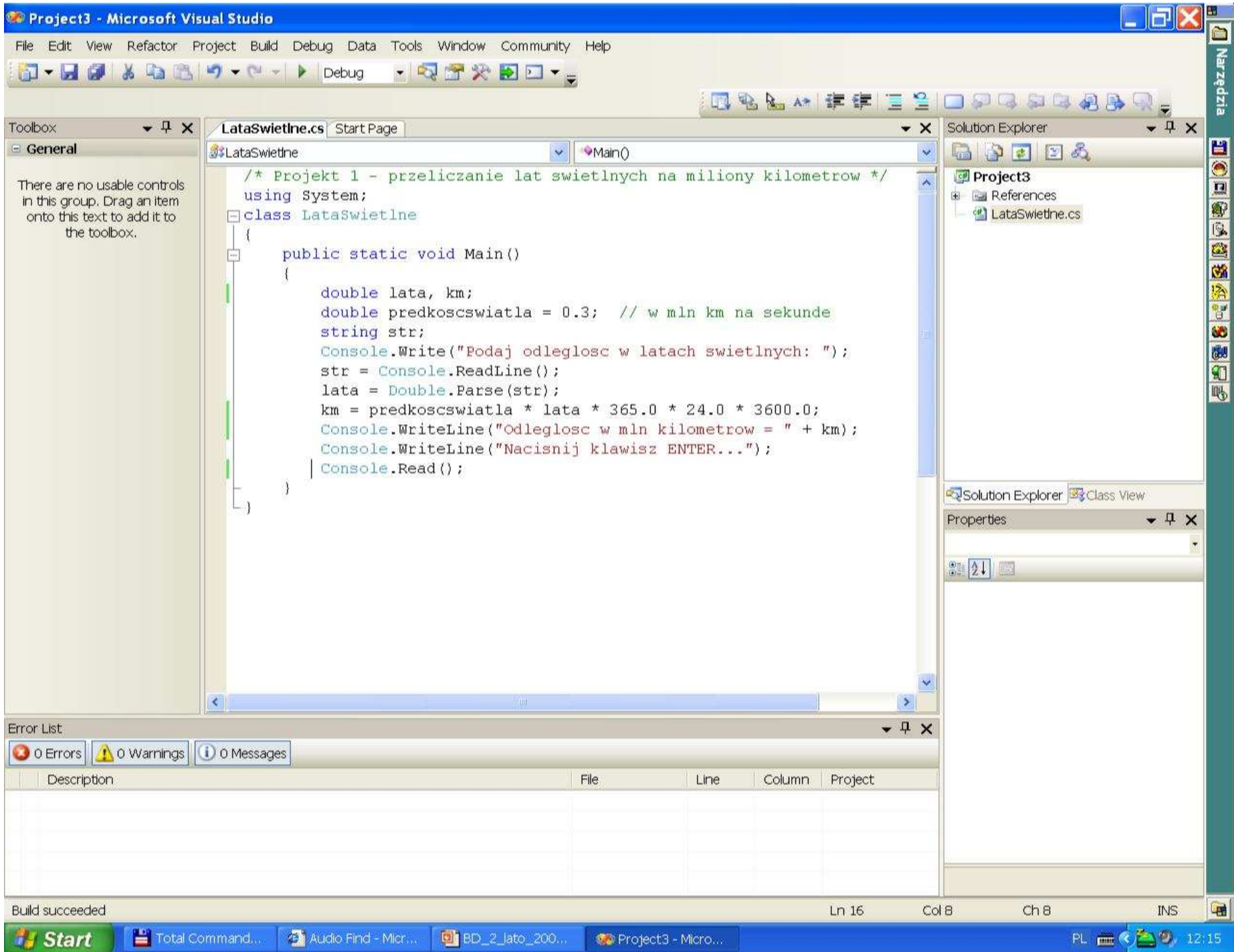
/* Projekt 1 - przeliczanie lat swietlnych na miliony kilometrow
*/
using System;
class LataSwietlne
{
    public static void Main()
    {
        double lata, km;
        double prędkośćświatła = 0.3; // w mln km na sekunde
        string str;
        Console.Write("Podaj odleglosc w latach swietlnych: ");
        str = Console.ReadLine();
        lata = Double.Parse(str);
        km = prędkośćświatła * lata * 365.0 * 24.0 * 3600.0;
        Console.WriteLine("Odległość w mln kilometrow = " + km);
        Console.WriteLine("Naciśnij klawisz ENTER...");
        Console.Read();
    }
}

```

```
Total Commander 6.51 - NOT REGISTERED
E:\Zajecia_lato_2006\Bazy_danych_2\Bazy_2_lato_2006\CSharp\ConsApp1\bin\Debug\Pr
Podaj odleglosc w latach swietlnych: 1230
Odleglosc w mln kilometrow = 11636784000
Nacisnij klawisz ENTER...
-
```

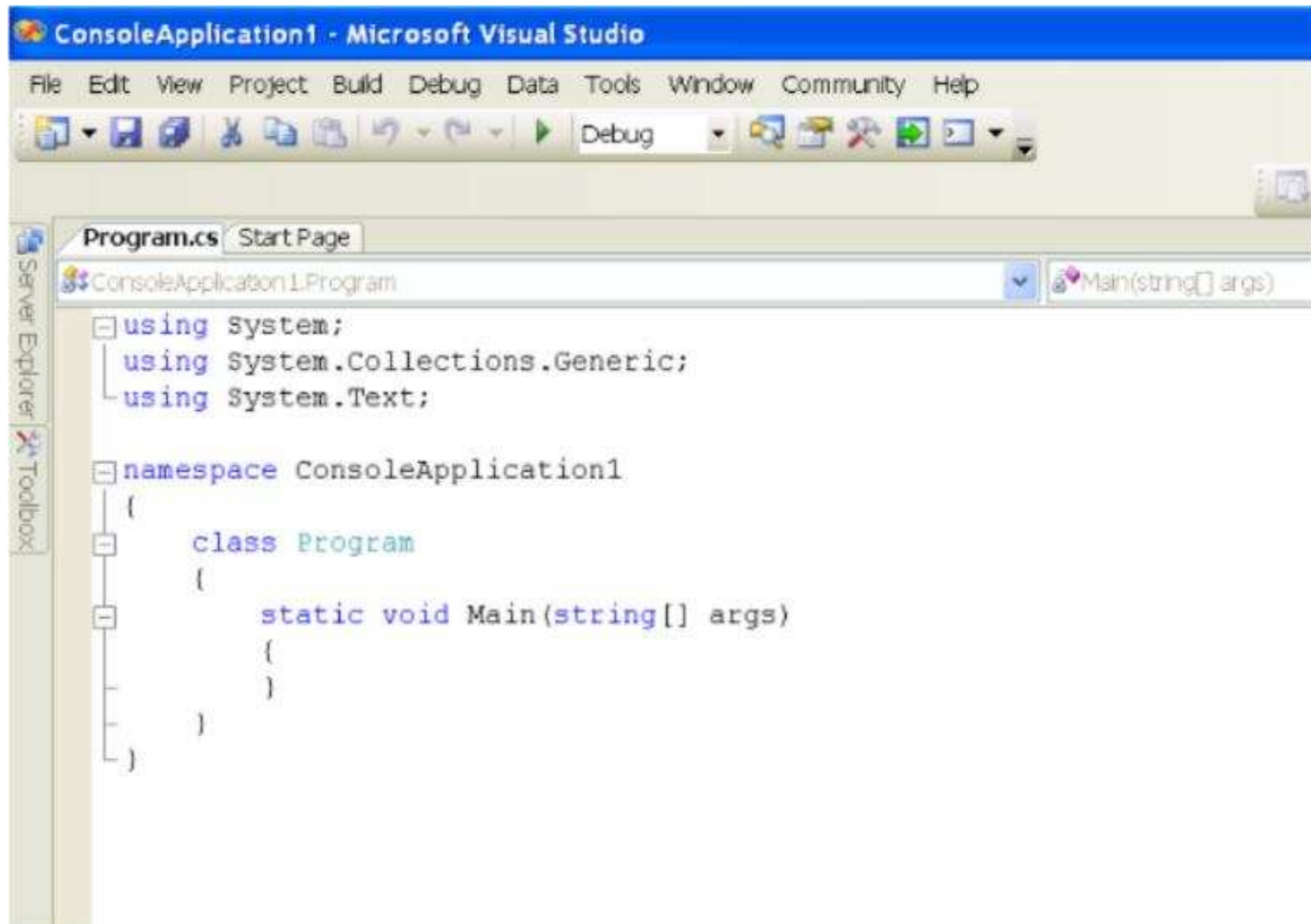
Środowisko IDE



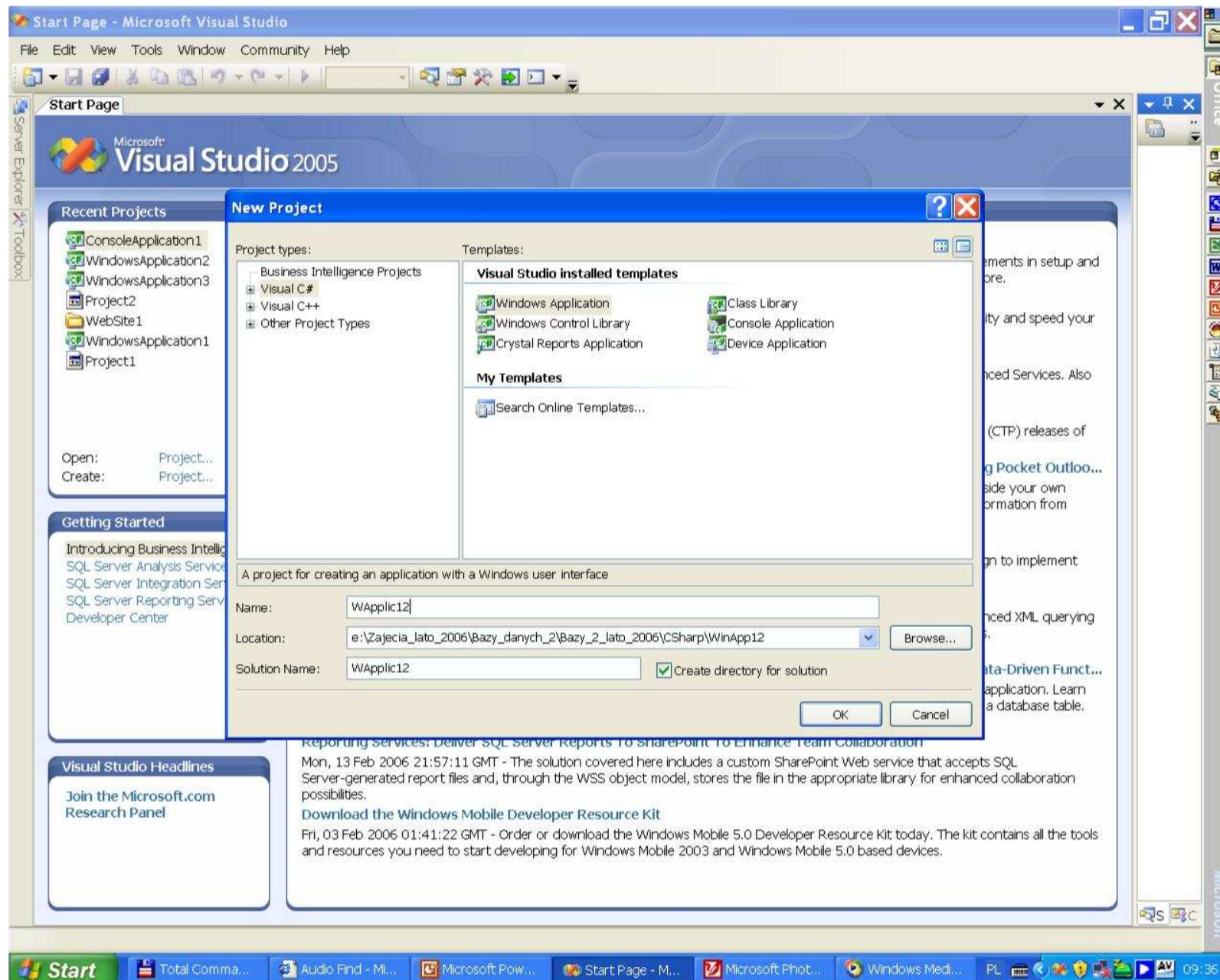


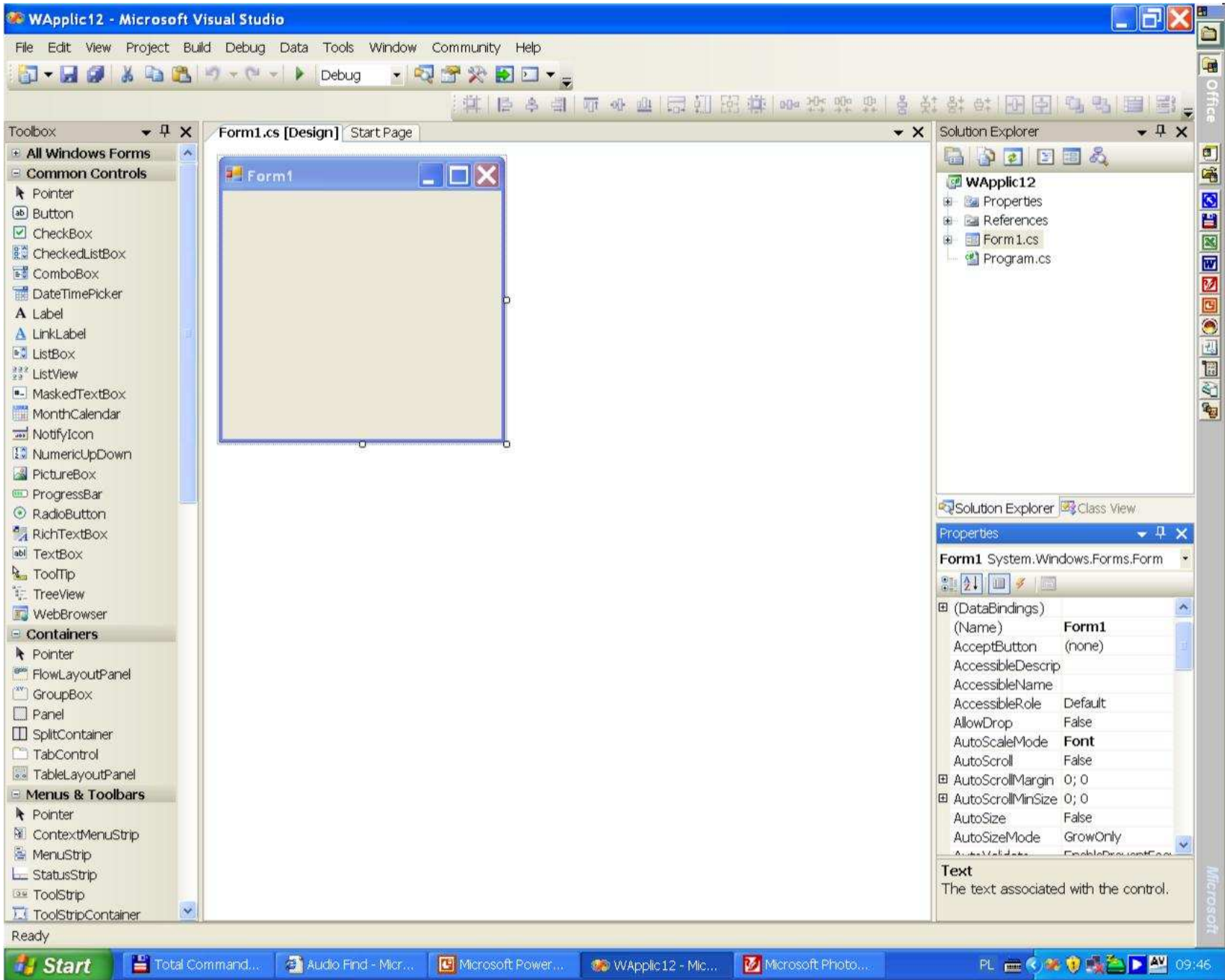
Narzędzia

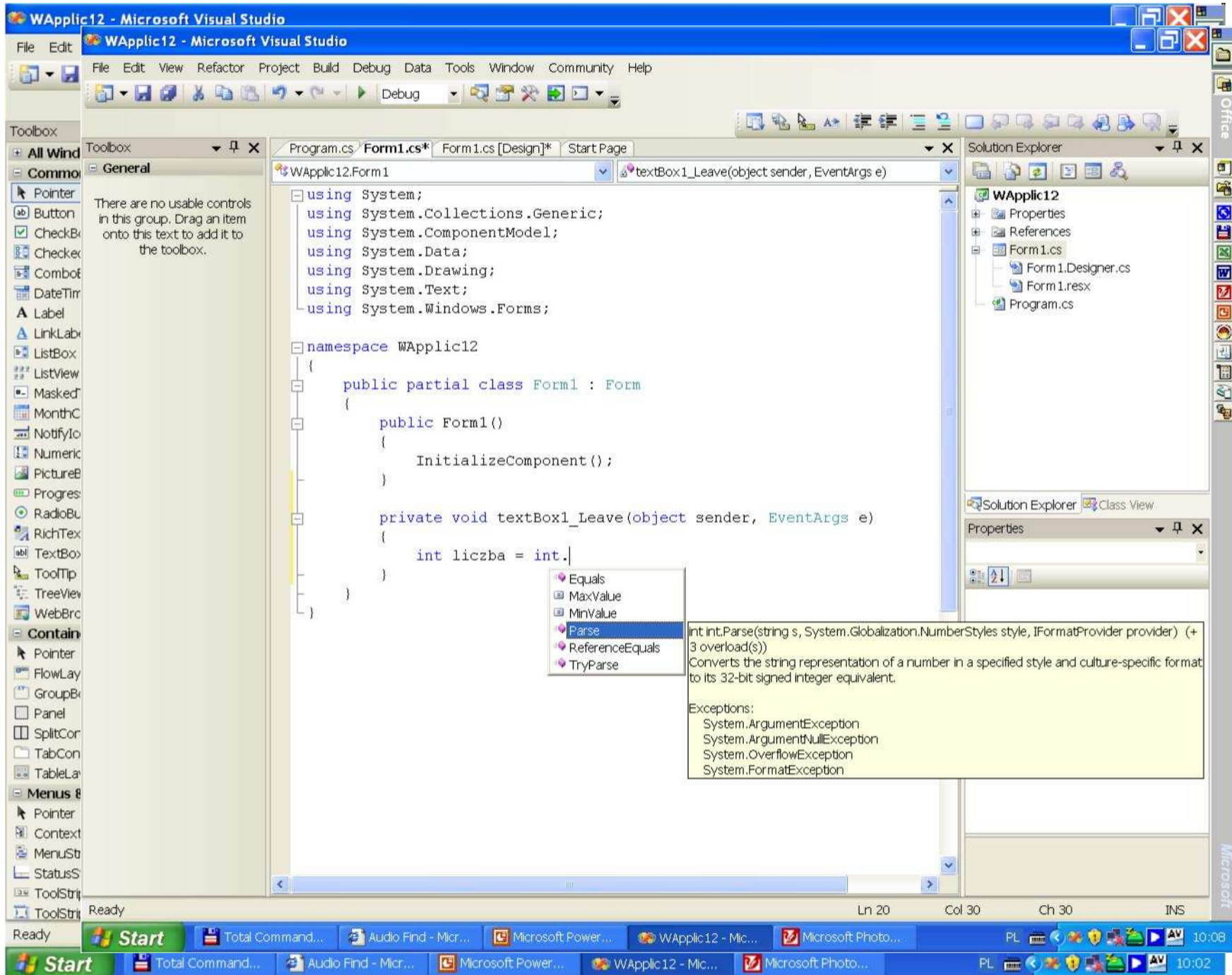
Aplikacja „konsolowa” - szablon

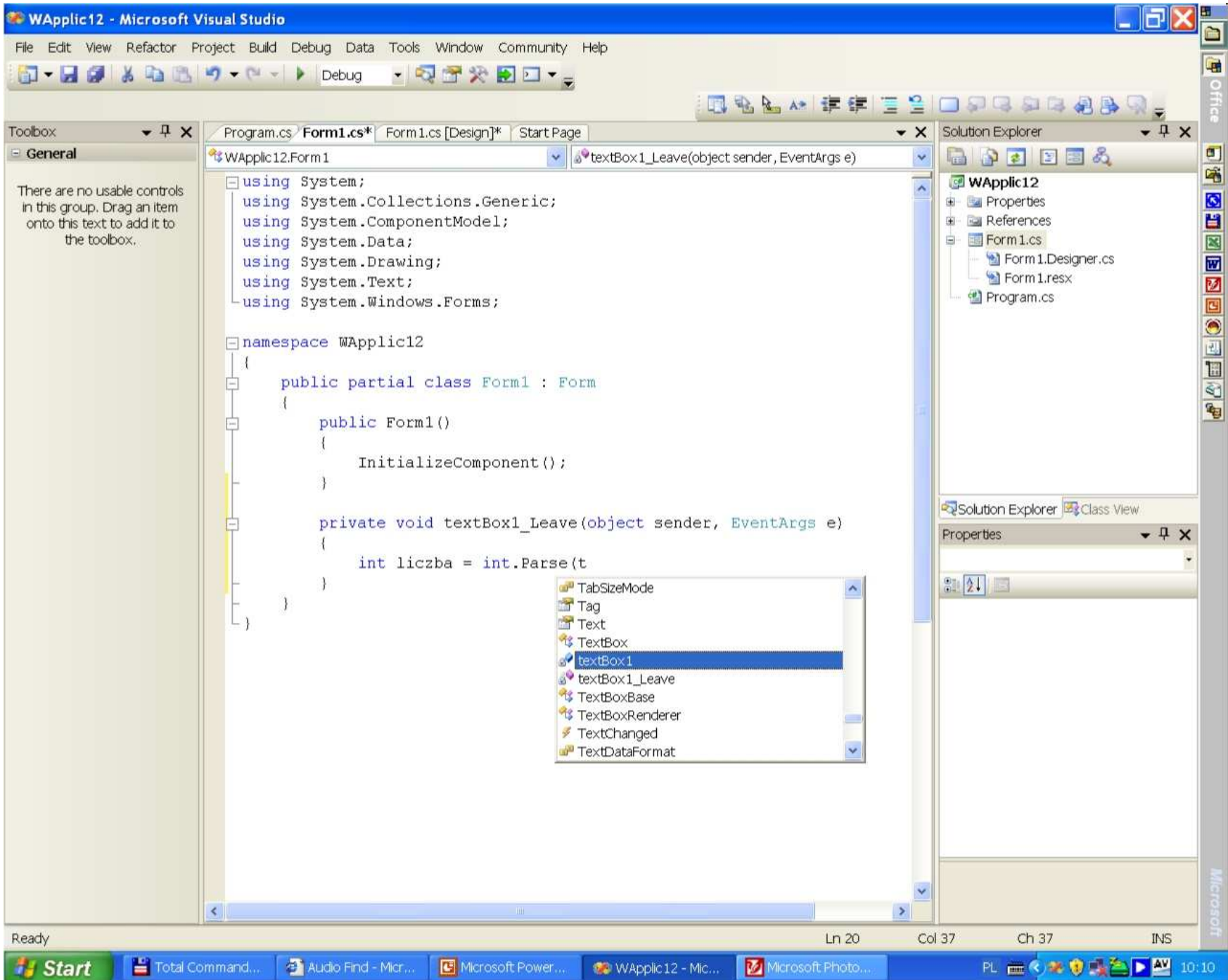


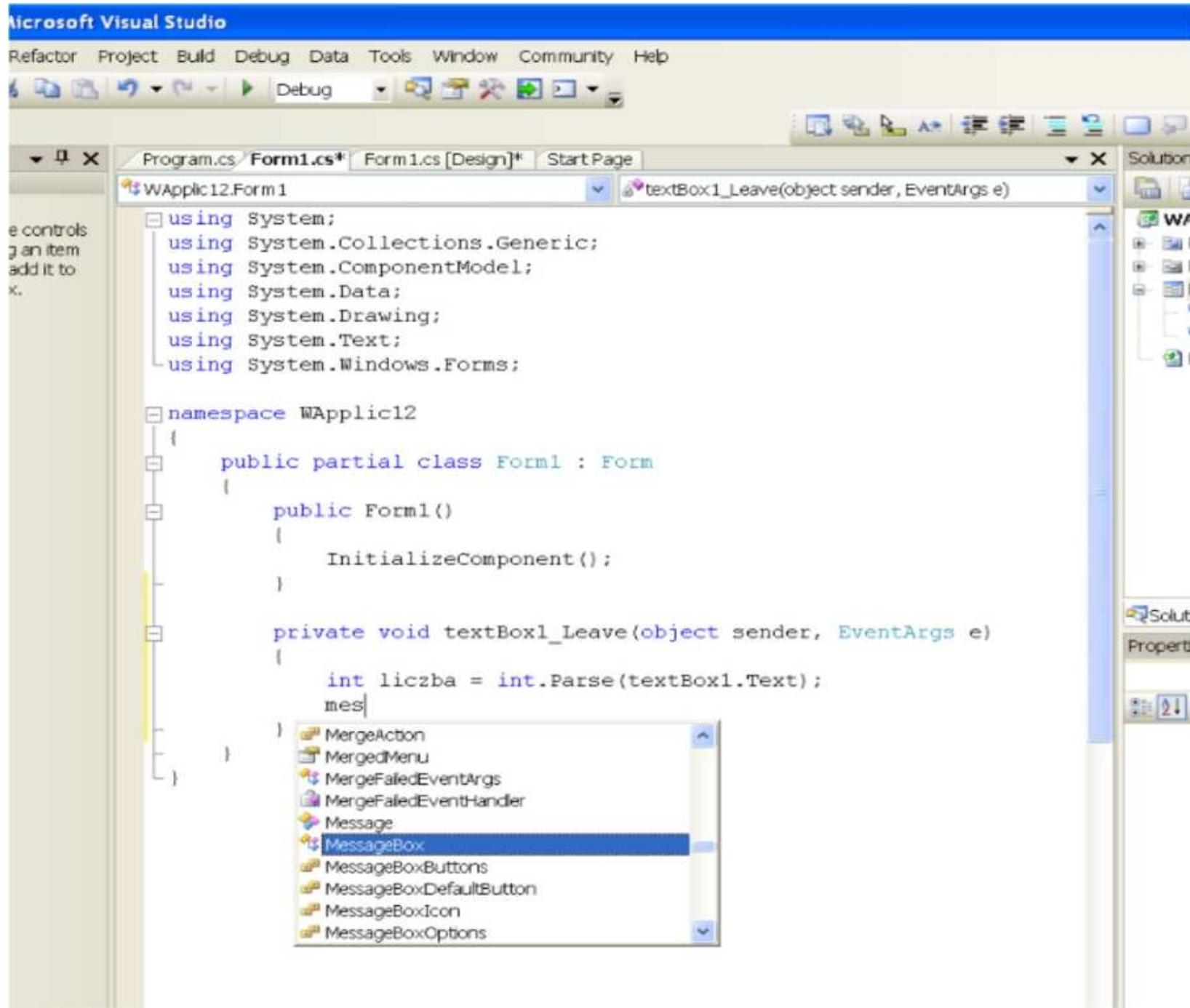
Wybór aplikacji „okienkowej”

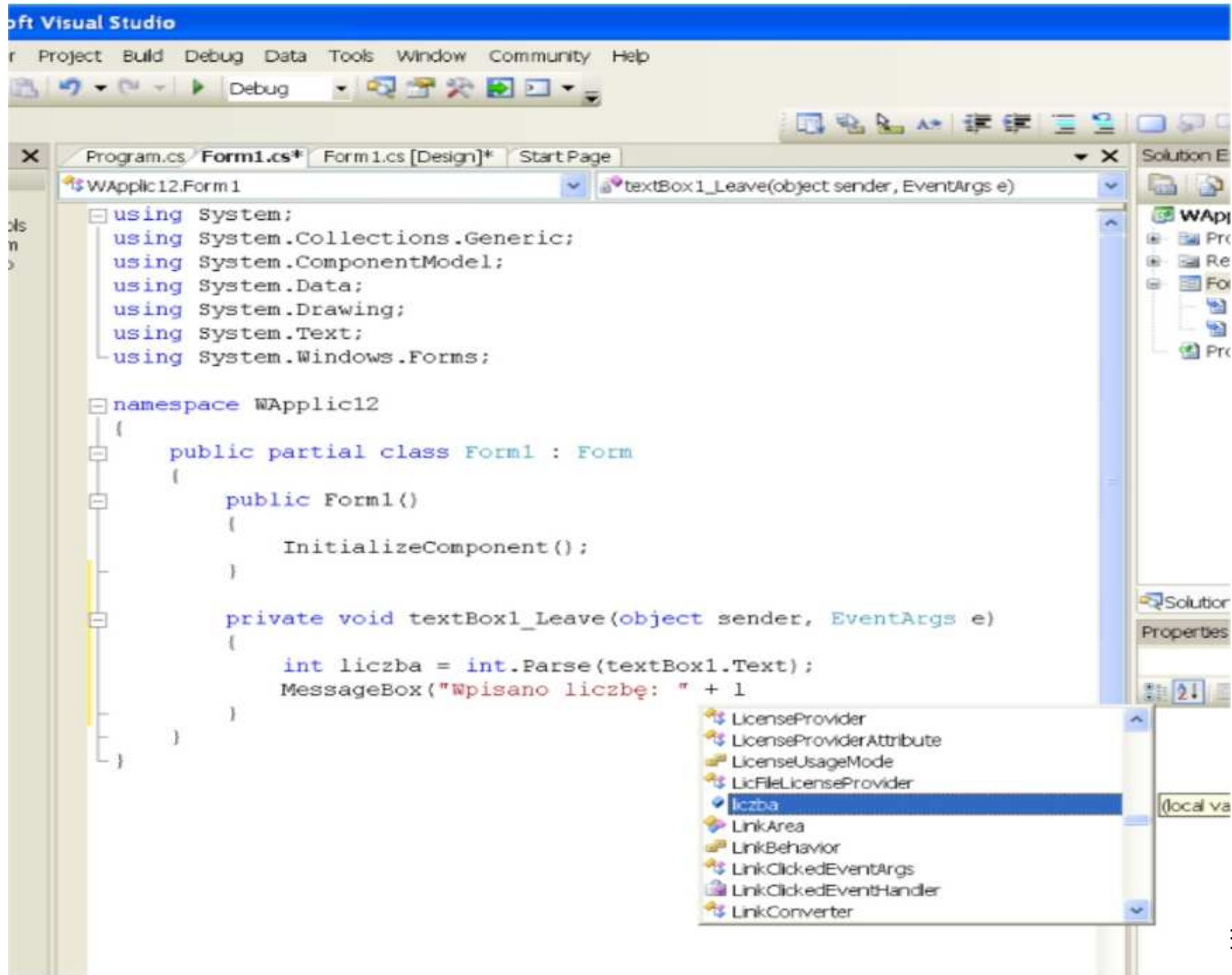


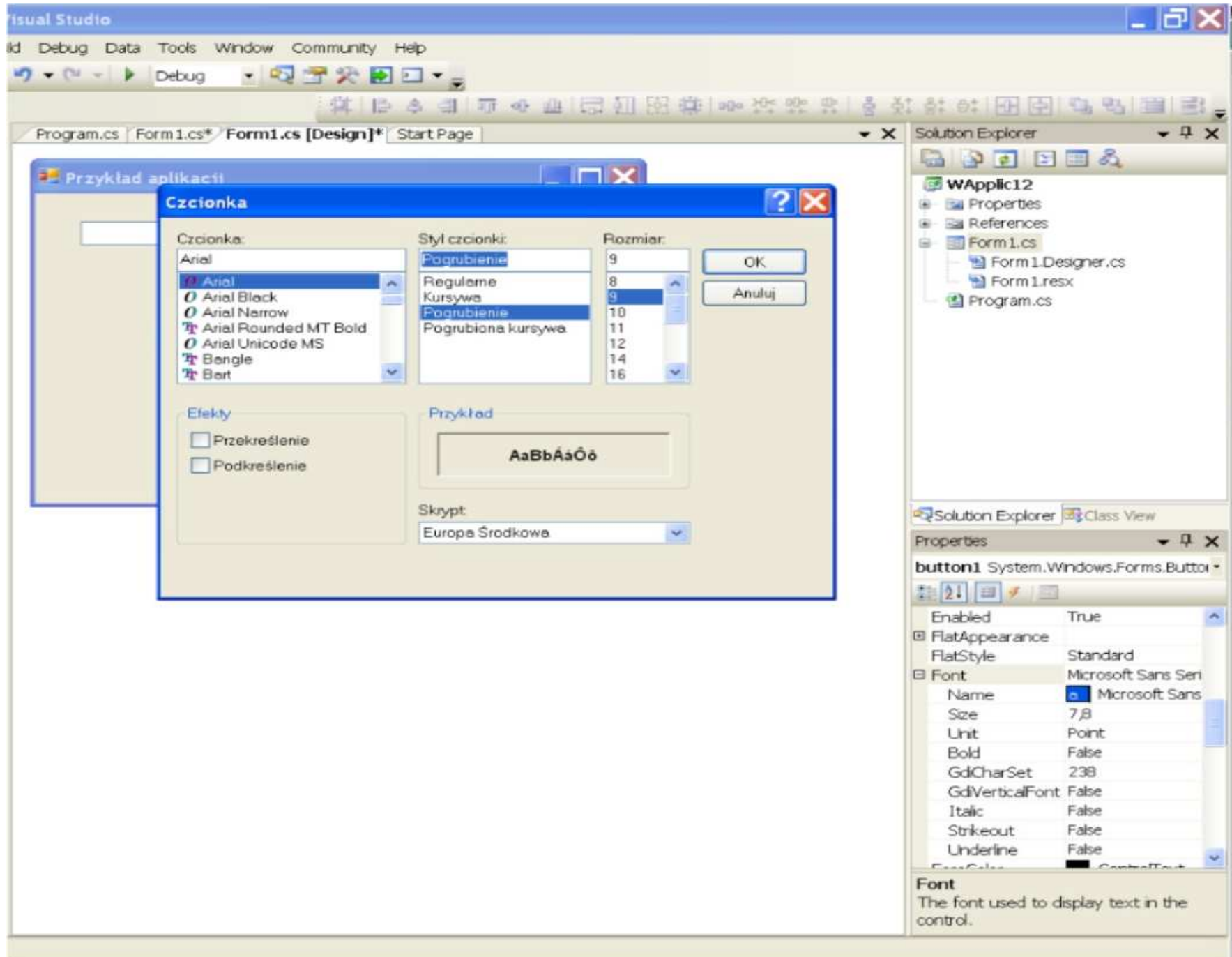


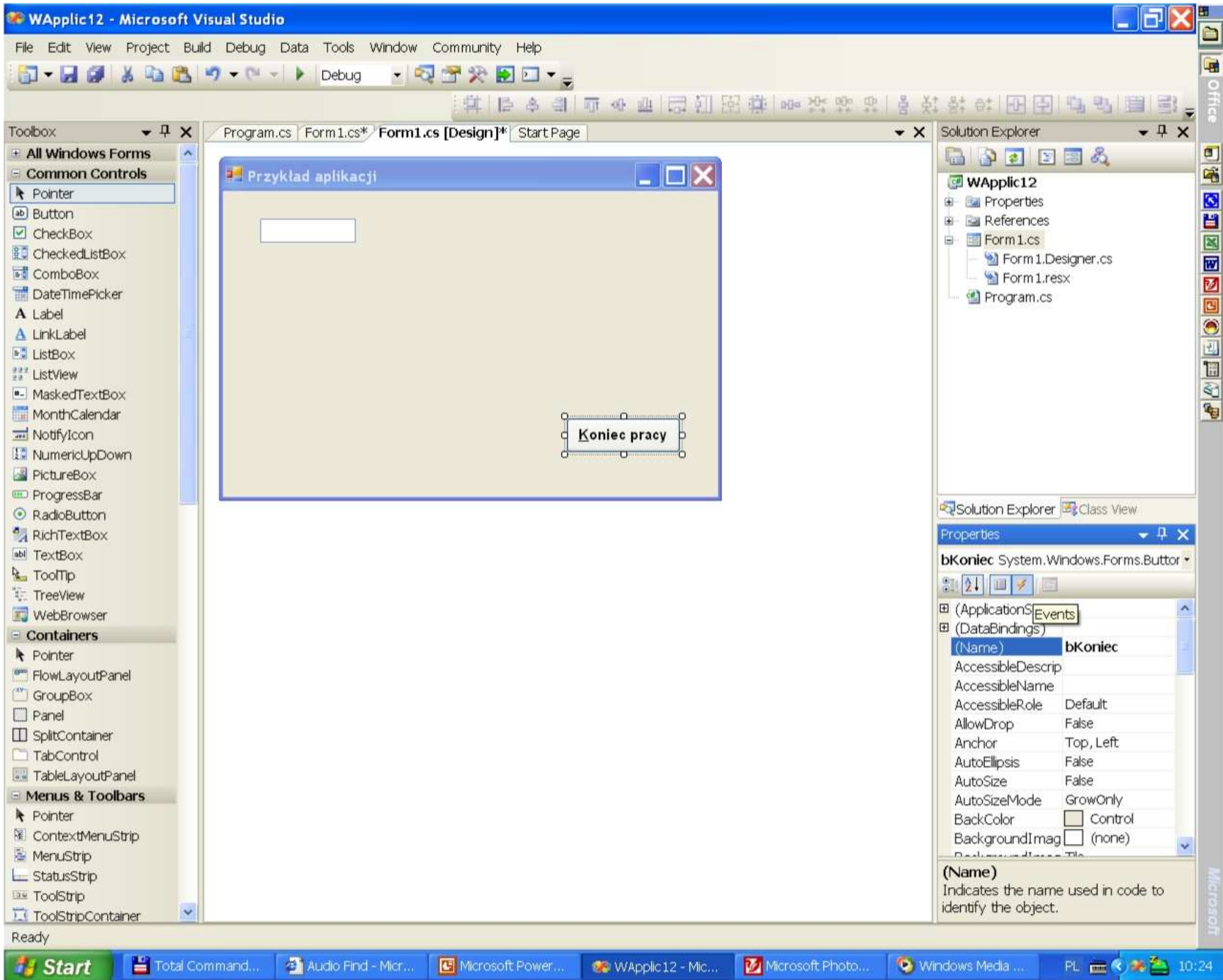


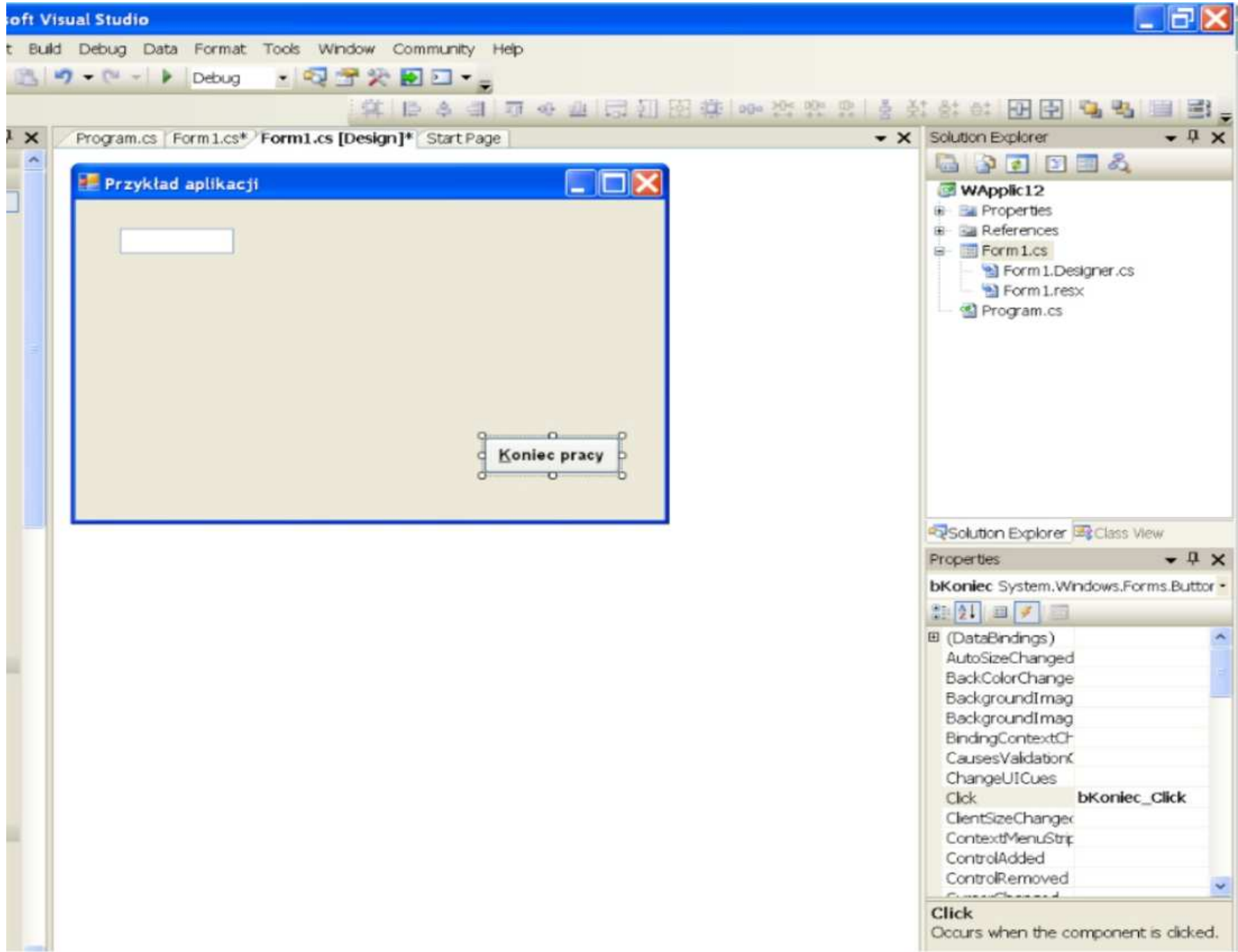












```
namespace WApplic12
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void textBox1_Leave(object sender, EventArgs e)
        {
            int liczba = int.Parse(textBox1.Text);
            MessageBox.Show("Wpisano liczbę: " + liczba);
        }

        private void bKoniec_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }
    }
}
```